

# Implementación de Hápticas en la interacción mano– pinza robótica.

Johann Hernan Correa Murillo

Proyecto de grado para optar al título de  
Ingeniero Electrónico

Universidad Tecnológica de Pereira  
Facultad de Ingenierías  
Ingeniería Electrónica  
Pereira, Colombia  
Julio 2018

# Implementación de Hápticas en la interacción mano– pinza robótica.

Por:  
Johann Hernan Correa Murillo  
Cód: 1111801868

Proyecto de grado para optar al título de  
Ingeniero Electrónico

Director:  
M.Sc. José Andrés Chaves Osorio  
Profesor del Programa Ingeniería Electrónica  
Ingeniero Electricista

Universidad Tecnológica de Pereira  
Facultad de Ingenierías  
Ingeniería Electrónica  
Pereira, Colombia  
Julio de 2018



## *Dedicatoria*

*Este proyecto de grado es dedicado a mis padres por ser el pilar fundamental de mi vida, por brindarme apoyo incondicional durante todos estos años de mi formación académica. Todo este trabajo ha sido posible gracias a ellos.*

*Johann Hernan Correa Murillo.*

## *Agradecimientos*

Primero que todo agradecerle a Dios ya que sin el este proyecto no habría sido posible, a mis familiares por todo el apoyo que me han brindado durante toda mi carrera, al M.Sc José Andrés Chaves Osorio que fue el encargado de dirigir este proyecto y a todo mis compañero del grupo de investigación robótica aplicada que en todo momento estuvieron dispuestos a brindarme su ayuda sin ningún compromiso. Son muchas las personas a las que debo agradecer por su ayuda y colaboración durante la ejecución de este proyecto y de toda mi carrera.

Johann Hernan Correa Murillo

# Índice general

<b>1. DESCRIPCIÓN DEL PROYECTO</b>	<b>8</b>
1.1. INTRODUCCIÓN	8
1.2. RESUMEN	8
1.3. OBJETIVOS	9
1.3.1. Objetivo general	9
1.3.2. Objetivos específicos	9
<b>2. MARCO REFERENCIAL</b>	<b>10</b>
2.1. MARCO HISTÓRICO	10
2.2. MARCO CONTEXTUAL	13
2.3. MARCO TEÓRICO-CONCEPTUAL	13
2.3.1. Concepto de transductor y sensor	13
2.3.2. Descripción de la IMU	13
2.3.2.1. Acelerómetro	14
2.3.2.2. Giroscopio	15
2.3.3. Flex sensor	15
2.3.4. Acondicionamiento de señal para sensores resistivos	15
2.3.5. Filtros	17
2.3.6. Comunicación inalámbrica	18
2.3.7. Definición de los conceptos de la programación estructurada y modular	18
<b>3. DESARROLLO DEL PROYECTO</b>	<b>19</b>
3.1. MÓDULO BLUETOOTH (BT) HC 05	19
3.2. PINZA ROBÓTICA	19
3.2.1. Montaje de la pinza robótica	19
3.2.2. Sensores en la pinza	22
3.2.2.1. Sensor de fuerza (FSR)	22
3.2.2.2. Instrumentación del sensor de fuerza (FSR)	22
3.2.3. Sensor de temperatura LM35	24
3.2.4. Adaptación de los sensores a la pinza robótica	26
3.2.5. Sistema de control de la pinza robótica	27
3.2.6. Circuito Electrónico para el control de la pinza robótica	28
3.3. GUANTE HÁPTICO	30
3.3.1. Sistema de control en el guante háptico	31
3.3.2. Arduino en el guante háptico	32
3.3.3. Control del motor DC y la resistencia calefactora en el guante	33
3.3.3.1. Motor DC	33
3.3.3.2. Resistencia calefactora	34
3.3.4. Lectura de la resistencia flexible	34
3.3.5. Uso del acelerómetro dentro del guante háptico	36
3.3.5.1. Implementación del acelerómetro de la IMU MPU 6050	36
3.3.5.2. Medición de los ángulos de inclinación	37

3.3.5.3.	Medición del ángulo de inclinación para los servos M4 y M3 . . . . .	37
3.3.5.4.	Medición del ángulo de inclinación para el servo M2 . . . . .	40
3.3.6.	Circuito Electrónico para el control del guante háptico . . . . .	42
<b>4.</b>	<b>PRUEBAS Y RESULTADOS</b>	<b>46</b>
4.1.	PRUEBAS DE FUNCIONAMIENTO DE LOS SERVO MOTORES . . . . .	46
4.2.	PRUEBAS HÁPTICAS DEL SISTEMA . . . . .	50
4.2.1.	Pruebas vibratorias . . . . .	50
4.2.2.	Pruebas térmicas . . . . .	54
<b>5.</b>	<b>CONCLUSIONES Y FUTUROS TRABAJOS</b>	<b>55</b>
5.1.	CONCLUSIONES . . . . .	55
5.2.	FUTUROS TRABAJOS . . . . .	56
<b>6.</b>	<b>PRESUPUESTO</b>	<b>57</b>
<b>A.</b>	<b>CÓDIGOS DESARROLLADOS</b>	<b>60</b>
A.1.	CÓDIGO IMPLEMENTADO PARA EL GUANTE . . . . .	60
A.2.	CÓDIGO IMPLEMENTADO PARA LA PINZA ROBÓTICA . . . . .	65

# Índice de figuras

2.1. Interfaz háptica GIRU diseñada usando SolidEdge® [1]. . . . .	11
2.2. Estructura cinemática del robot LapBot [2]. . . . .	11
2.3. Prototipo del sistema háptico utilizando realimentación con aire para un simulador de cometa tipo ala delta [3]. . . . .	12
2.4. CyberGrasp [4]. . . . .	12
2.5. CyberGlove [9] . . . . .	13
2.6. Drone Phantom 3 pro incorpora una IMU para su funcionamiento [5]. . . . .	14
2.7. IMU MPU 6050. . . . .	14
2.8. Flex sensor (resistencia flexible). . . . .	15
2.9. Circuito para la implementación del método de deflexión. . . . .	16
2.10. Circuito para la implementación del método de doble lectura. . . . .	16
2.11. Circuito para la implementación del método del divisor de voltaje. . . . .	17
2.12. configuración circuital de un filtro a) pasivo, b) activo [6][7]. . . . .	17
2.13. Comunicación inalámbrica [8]. . . . .	18
3.1. Módulo BT HC05 . . . . .	19
3.2. Pinza robótica. . . . .	20
3.3. Servomotor MG966R. . . . .	21
3.4. Pinza robótica con los topes mecánicos. . . . .	21
3.5. Sensor de fuerza FSR 402. . . . .	22
3.6. Código para la lectura del puerto análogo-digital del arduino. . . . .	24
3.7. Sensor de temperatura LM35. . . . .	25
3.8. Adaptador para la pinza con los sensores de fuerza. . . . .	26
3.9. Adaptador para la pinza con los sensores de temperatura. . . . .	26
3.10. Pinza robótica con las piezas y los sensores montados. . . . .	27
3.11. Función de control para el servo. . . . .	27
3.12. Esquemático del circuito de control de la pinza robótica. . . . .	28
3.13. Board del circuito de control de la pinza robótica. . . . .	28
3.14. PCB del circuito de control de la pinza robótica. . . . .	29
3.15. Caja para la circuitería de la pinza. . . . .	29
3.16. Montaje completo de la pinza robótica . . . . .	30
3.17. Motor DC incorporado en el guante háptico. . . . .	30
3.18. Resistencia calefactora incorporado en el guante háptico. . . . .	31
3.19. Guante háptico implementado en el proyecto. . . . .	32
3.20. Arduino nano V3.0. . . . .	32
3.21. Función analogwrite de arduino. . . . .	34
3.22. Código de la función map() de arduino. . . . .	36
3.23. Módulo MPU 6050 dentro de la pieza. . . . .	36
3.24. Posición de reposo de la IMU MPU 6050 con su componente gravitatoria en Z. . . . .	37
3.25. IMU MPU 6059 rotada 90° con su componente gravitatoria en X. . . . .	37
3.26. Posición de reposo del operador con las componentes gravitatorias en la muñeca. . . . .	38
3.27. Posición a 45° de la muñeca. . . . .	39
3.28. Inclinación a 45° de la muñeca en los ejes X y Y. . . . .	40

3.29. Posición de reposo del operador con las componentes gravitatorias en el bicep. . . . .	41
3.30. Movimiento del bicep a $90^\circ$ . . . . .	42
3.31. Esquemático del circuito de control del guante. . . . .	43
3.32. Board del circuito para el guante. . . . .	43
3.33. Montaje completo del circuito de control del guante. . . . .	44
3.34. a) caja con la PCB, b) montaje de la caja con la PCB y la batería. . . . .	44
3.35. Sistema de control del guante completado. . . . .	45
4.1. Posición inicial del sistema. . . . .	46
4.2. Prueba del movimiento vertical de la muñeca a $-45^\circ$ (M3). . . . .	47
4.3. Prueba del movimiento vertical de la muñeca a $90^\circ$ (M3). . . . .	47
4.4. Prueba del movimiento rotatorio de la muñeca en sentido horario(M4). . . . .	48
4.5. Prueba del movimiento vertical del bicep (M2). . . . .	48
4.6. Prueba del movimiento rotatorio de la base en el sentido antihorario (M1). . . . .	49
4.7. Prueba del movimiento rotatorio de la base en el sentido horario (M1). . . . .	49
4.8. Prototipo de medición de fuerza. . . . .	50
4.9. Pruebas de fuerza realizadas con el prototipo de medición de fuerza. . . . .	51
4.10. Grafica de Fuerza vs Duty. . . . .	52
4.11. Grafica de Fuerza vs Duty con offset. . . . .	53
4.12. Pruebas vibratorias del sistema hápticas. . . . .	53
4.13. Temperatura presente en el guante cuando no se detecta ningún objeto caliente. . . . .	54
4.14. a) temperatura presente en la pinza, b) temperatura presente en el guante. . . . .	54

# Índice de cuadros

3.1. Elementos de la pinza robótica. . . . .	20
3.2. Valores resistivos máximos y mínimos de los sensores de fuerza. . . . .	23
3.3. Valores máximos y mínimos de voltajes entregados por los sensores. . . . .	23
3.4. Valores digitales. . . . .	24
3.5. Rangos de operación de los sensores expresados en °C. . . . .	25
3.6. Rangos de operación de los sensores expresados en voltios (V). . . . .	25
3.7. Rangos de operación de los sensores digitalizados. . . . .	25
3.8. Especificaciones del arduino nano V3.0. . . . .	33
3.9. Valores escalados para los motores DC. . . . .	34
3.10. . Rangos de operación de los sensores expresado en ohmios ( $\Omega$ ). . . . .	35
3.11. Rangos de operación de los sensores expresado en voltios (V). . . . .	35
3.12. Rangos de operación de los sensores digitalizados. . . . .	35
4.1. Relación entre fuerza, voltaje y Duty. . . . .	51
4.2. Relación entre fuerza, voltaje y Duty con offset incluido. . . . .	52
6.1. presupuesto. . . . .	57

# Capítulo 1

## DESCRIPCIÓN DEL PROYECTO

### 1.1 INTRODUCCIÓN

La háptica es la ciencia encargada del estudio de las sensaciones y del tacto en el ámbito de la robótica. El sentido del tacto es uno de los más importantes para los seres humanos, a través de él se puede obtener información de texturas y superficies, además advierte al cuerpo cuando se encuentra en peligro; es importante resaltar que el tacto es el primer sentido que se desarrolla en el feto. Es aquí donde las tecnologías hápticas aprovechan el sentido del tacto para construir herramientas las cuales puedan generar estimulaciones y ser utilizadas en diferentes campos [9].

En la actualidad existen diferentes tipos de dispositivos hápticos la mayor parte de ellos enfocados a los videojuegos, pero también hay dispositivos mucho más robustos como por ejemplo, **Butterfly Haptics** [10]. Se pueden encontrar tres grandes ramas para el desarrollo de las tecnologías hápticas: la robótica humanoide con uno de los diseños más impactantes que es **Shadow Hand**, en la medicina con simulación quirúrgica y en las artes [11].

**Shadow Hand**, es una mano humanoide que ha sido desarrollado por **The Shadow Robot Company en Londres**, ésta es comparable a la mano de un adulto promedio en tamaño y forma, además consta con 20 grados de libertad accionados, dándole mayor movilidad. Este humanoide tiene incorporados diferentes tipos de sensores como, por ejemplo, sensores de posición, sensores de fuerza absoluta, sensores de contacto ultrasensibles en las yemas de los dedos, los cuales reproducen el sentido del tacto de un humano, brindando información exacta del entorno en el cual la **Shadow Hand** esta interactuando [12].

En esta investigación se trabajará en el diseño e implementación de un guante háptico, el cual controlara remotamente una pinza robótica. Esta permitirá al operador percibir la temperatura a la cual se encuentra el objeto que la pinza sujeta por medio del uso de una resistencia calefactora que estará instalada en el guante, así mismo también podrá percibir el contacto con el objeto por medio de pequeñas vibraciones en sus dedos.

### 1.2 RESUMEN

Las hápticas hacen parte del conjunto de tecnologías que están revolucionando el mundo. En la actualidad, estas tecnologías se están aplicando en la medicina, la robótica, los videojuegos entre otras, siendo este último campo en el cual ha tenido mucho impacto. Las hápticas permiten al usuario percibir por medio del tacto objetos que no existen (virtuales) u objetos remotos y de esta forma obtener una interacción entre el operador y un objeto “no real”. En este trabajo se aplicarán las tecnologías hápticas para diseñar y construir un guante háptico, con sensores incorporados para el control remoto de una pinza robótica, la cual le permitirá al operador obtener una realimentación vibratoria y térmica de los objetos que agarre la pinza, con el objetivo de manipular objetos peligrosos y brindar información táctil al operador del objeto; la pinza cuenta con 5 grados de libertad permitiéndole tener un buen rango de operación.



## **1.3 OBJETIVOS**

### **1.3.1. Objetivo general**

Diseñar y construir un guante háptico con sensores incorporados para generar la interacción háptica mano-pinza robótica.

### **1.3.2. Objetivos específicos**

- Diseñar el guante háptico que permita controlar remotamente una pinza robótica, generando dos realimentaciones una vibratoria y una térmica asociada al objeto que se está manipulando.
- Desarrollar el circuito electrónico de la pinza robótica para que pueda ser controlada remotamente y a su vez captar la información del objeto que se está manipulando.
- Implementar el software encargado de la comunicación inalámbrica y el control de los sistemas.

## Capítulo 2

# MARCO REFERENCIAL

### 2.1 MARCO HISTÓRICO

El hombre siempre se ha dado a la tarea de diseñar y construir herramientas que le permitan ejecutar diversas actividades de manera más sencilla muchas de estas ponen en riesgo su vida por lo cual también se ha planteado diseñar y construir máquinas que le permitan realizar estas tareas evitando poner en riesgo su vida, cabe resaltar que actualmente existen diversas tareas que ponen en riesgo la vida del ser humano como por ejemplo el manejo de químicos altamente tóxicos, el manejo explosivos como minas antipersonal, etc; a partir de esta necesidad nacen algunos dispositivos teleoperados como **ANGUS** para minimizar estos riesgos. **ANGUS** es un robot para desminado humanitario el cual se desarrolló en el grupo de investigación robótica aplicada de la Universidad Tecnológica de Pereira, este es una plataforma móvil robusta el cual es capaz de detectar materiales metálicos utilizados en las minas antipersonales a una profundidad de 18cm [13]. Con el pasar del tiempo nace la tecnología háptica la cual se empieza a ligar con los instrumentos teleoperados creando dispositivos los cuales pueden ser operados remotamente y a su vez obtener una realimentación sensorial.

A continuación se mencionarán algunos trabajos relacionados con las tecnologías hápticas que hacen parte del estado del arte de esta investigación.

En la Universidad del Cauca en Popayán los estudiantes Jimmer William Genoy Muñoz, Luis Felipe Rodríguez Ortiz y Sergio Alexander Salinas diseñaron un dispositivo háptico para aplicaciones biomédicas para medicina quirúrgica en donde el operador siente y manipula objetos simulados, este dispositivo cuenta con cuatro grados de libertad permitiendo al operador posicionar y orientar el efector final en el ambiente de trabajo [1].



Figura 2.1: Interfaz háptica GIRU diseñada usando SolidEdge® [1].

En la Universidad del Cauca en Popayán se desarrolló un robot (LapBot) para cirugías laparoscópicas en un simulador quirúrgico, el cual cuenta con 9 grados de libertad los cuales 6 de ellos son articulaciones activas que le permiten al robot posicionar el instrumento quirúrgico en un espacio tridimensional, los otros tres grados de libertad se consideran pasivos, estos grados de libertad garantizan que el robot sea seguro y no le cause una lesión al paciente por algún movimiento brusco a la hora de realizar la incisión, el sistema tiene incorporada la detección de colisiones haciendo uso de motores físicos [2].

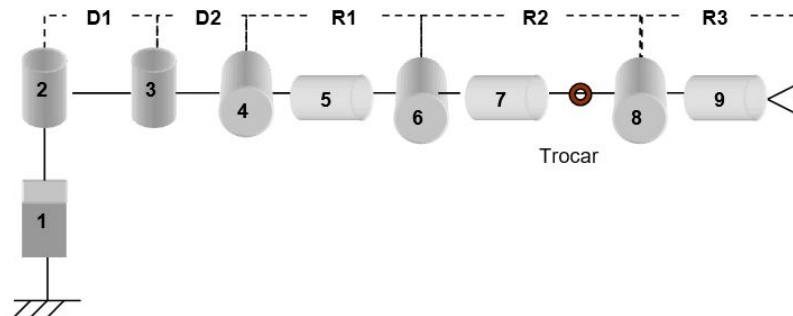


Figura 2.2: Estructura cinemática del robot LapBot [2].

En la universidad Militar Nueva Granada de Bogotá el estudiante Christian David Paternina diseño y construyó un prototipo de un simulador con realidad virtual de vuelo de cometa tipo ala delta el cual tiene una respuesta háptica de aire, es decir el usuario realiza un vuelo sobre un terreno virtual y puede sentir las ventiscas que se generan en el entorno virtual por medio de un arreglo de pequeños ventiladores los cuales son los encargados de generar la respuesta háptica de aire en la piel del usuario [3].

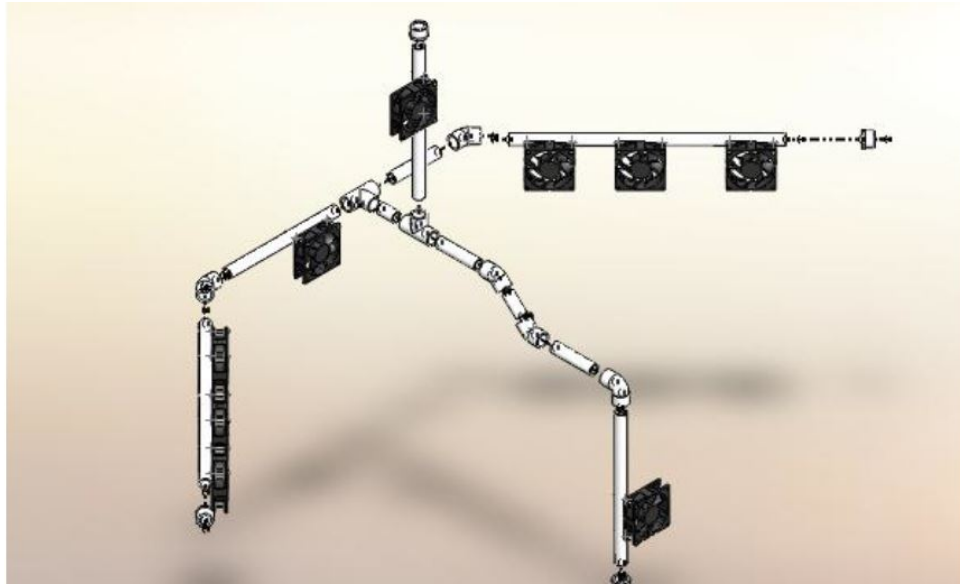


Figura 2.3: Prototipo del sistema háptico utilizando realimentación con aire para un simulador de cometa tipo ala delta [3].

En el mercado se encuentra un guante háptico llamado Cybergrasp fabricado por Immersio Co, el cual consiste en un sistema de realimentación de fuerzas para los dedos y la mano. Este guante le permite al usuario “estar dentro de una computadora” y captar o sentir objetos generados a través de esta. El CyberGrasp es un exoesqueleto liviano y resistente a la fuerza el cual agrega realimentación de fuerzas resistivas a cada dedo. Las fuerzas de agarre son producidas por una red de tendones dirigidos a las yemas de los dedos a través del exoesqueleto. Este guante cuenta con cinco actuadores, uno para cada dedo, los cuales se pueden programar individualmente para evitar que los dedos del operador deformen el objeto virtual [4].



Figura 2.4: CyberGrasp [4].

También se encuentra el guante háptico CyberGlove desarrollado por Immersio Co, este es un guante el cual tiene incorporados pequeños estimuladores vibrotáctiles en cada uno de sus dedos. Cada uno de estos estimuladores se pueden programar individualmente para modificar la intensidad de la sensación táctil [4].



Figura 2.5: CyberGlove [9]

## 2.2 MARCO CONTEXTUAL

El desarrollo del guante háptico para la interacción mano- pinza robótica se llevó acabo en la ciudad de Pereira, Colombia, en la Universidad Tecnológica de Pereira, en el programa ingeniería electrónica, en las instalaciones de investigación robótica aplicada.

## 2.3 MARCO TEÓRICO-CONCEPTUAL

Para el desarrollo del presente proyecto se deben de tener en cuenta conceptos que permitan comprender mejor los pasos que facilitaran la consecución del mismo.

### 2.3.1. Concepto de transductor y sensor

- **Transductor:** Un transductor es un dispositivo que puede convertir un tipo de energía en otra, como por ejemplo térmica a eléctrica, eléctrica a mecánica, ondas sonoras a eléctrica, etc [14].
- **Sensor:** Los sensores son dispositivos que hacen parte de los transductores, se define como sensor a todo aquel dispositivos que es sensible a una magnitud física convirtiendo esta a una señal eléctrica [15].

### 2.3.2. Descripción de la IMU

Una unidad de medición inercial o IMU es un dispositivo el cual puede tener integrado varios sensores, normalmente integra acelerómetros y giroscopios, este dispositivo es utilizado en sistemas de navegación como aviones, barcos, en dispositivos no tripulados como drones, misiles, en los accesorios para los videojuegos entre otros, en la figura 2.6 se observa un dispositivo no tripulado (drone) [16][17].



Figura 2.6: Drone Phantom 3 pro incorpora una IMU para su funcionamiento [5].

El módulo MPU 6050 contiene 6 grados de libertad (DoF), un acelerómetro de 3 ejes y un giroscopio de 3 ejes de alta precisión. Este dispositivo para poder medir la aceleración en los tres ejes posee tres sensores de las mismas características donde cada sensor está ubicado ortogonalmente uno con respecto al otro, de igual manera hay tres sensores para medir la velocidad angular en los tres ejes [16].

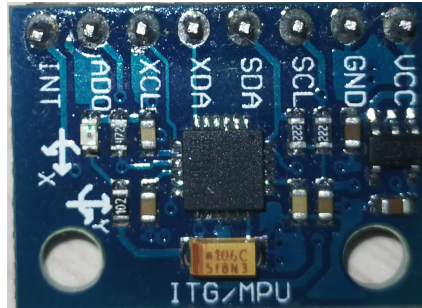


Figura 2.7: IMU MPU 6050.

#### 2.3.2.1. Acelerómetro

Un acelerómetro es un dispositivo que mide las vibraciones o aceleraciones del movimiento que experimenta un cuerpo. La aceleración es la razón de cambio en la velocidad con respecto al tiempo.

$$a = \frac{dv}{dt} \quad (2.1)$$

De acuerdo a la segunda ley de Newton dice que un cuerpo con masa constante la aceleración es directamente proporcional a la fuerza que actúa sobre el cuerpo.

$$a = \frac{F}{m} \quad (2.2)$$

Este criterio es utilizado en los acelerómetros o IMU para medir la aceleración, en donde los acelerómetros internamente contienen un MEMS (Micro Electro Mechanical Systems) que de forma análoga a un sistema masa resorte que permite medir la aceleración, normalmente la aceleración se mide en  $m/s^2$  este módulo utiliza la fuerza g, en donde 1g equivale a  $9,8m/s^2$  [16].

#### 2.3.2.2. Giroscopio

La velocidad angular es que tan rápido un objeto gira alrededor de su propio eje, normalmente las unidades con las cuales son medidos estos instrumentos son grados por segundo ( $^{\circ}/s$ ).

$$w = \frac{d\Theta}{dt} \quad (2.3)$$

El giroscopio incluido en el módulo MPU 6050 utiliza un MEMS (Micro Electro Mechanical Systems) para medir la velocidad angular utilizando el efecto Coriolis [16].

#### 2.3.3. Flex sensor

Una resistencia flexible es un elemento electrónico de tipo resistivo el cual varía su resistencia a medida que se realiza una flexión sobre el, este tipo de sensores para generar algún tipo señal eléctrica como voltaje o corriente necesita de una alimentación externa que le proporcione este tipo de señal [18].

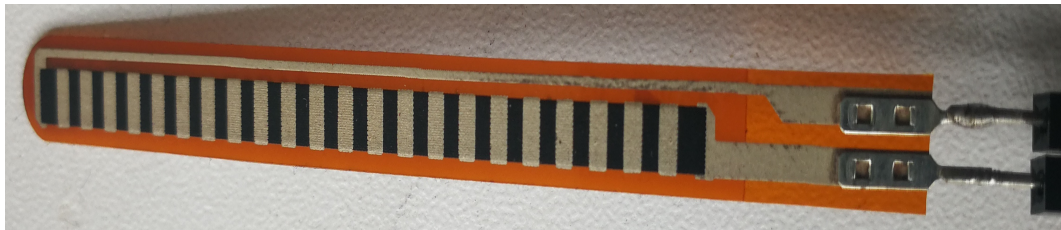


Figura 2.8: Flex sensor (resistencia flexible).

#### 2.3.4. Acondicionamiento de señal para sensores resistivos

Para el acondicionamiento de los sensores tipo resistivo existen varios métodos, los cuales se conocen como los métodos de deflexión y métodos de comparación [19].

Uno de los métodos por deflexión más simple consiste en alimentar directamente el sensor con una fuente de voltaje o corriente y medir la caída de tensión o voltaje que presenta, este método tiene un problema, al realizar una medición hay un cambio aproximadamente del 1 % lo que supone medir valor de corriente o voltaje muy pequeños.

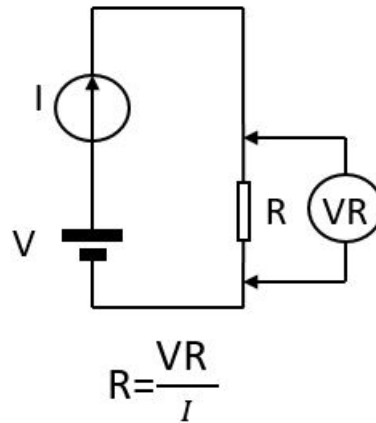


Figura 2.9: Circuito para la implementación del método de deflexión.

El método de doble lectura utiliza una resistencia la cual su valor es conocido en serie con el sensor, con el objetivo de determinar la resistencia del sensor a partir del voltaje en la resistencia conocida y el voltaje en el sensor.

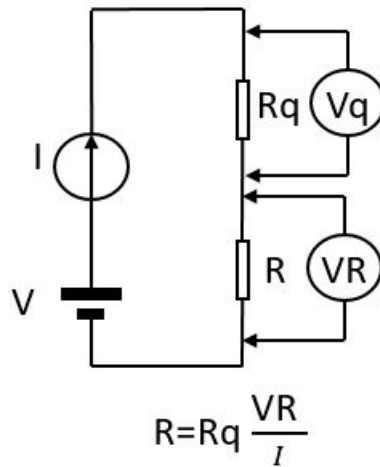


Figura 2.10: Circuito para la implementación del método de doble lectura.

El método del divisor de voltaje toma la tensión que existe en el sensor para calcular el valor de la resistencia del sensor, de igual forma que el método de doble lectura se utiliza una resistencia en serie con el sensor la cual se conoce su valor.



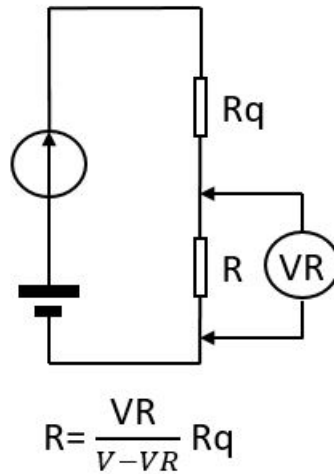


Figura 2.11: Circuito para la implementación del método del divisor de voltaje.

### 2.3.5. Filtros

El ruido eléctrico es toda aquella interferencia o perturbación que se presenta en las señales alterando la información de la señal. El ruido eléctrico es uno de los peores enemigos de las señales eléctricas, existen diversos métodos para eliminar un porcentaje considerable del ruido ya que este no puede ser eliminado en su totalidad. Para eliminar el ruido se utilizan filtros pasivos o activos en donde los filtros activos están conformados generalmente por bobinas, resistencias y condensadores, los filtros pasivos son construidos con amplificadores operacionales, resistencias y condensadores, en donde la función de estos filtros es eliminar frecuencias las cuales no estén en el rango de operación y de esta forma evitar señales no deseadas [6][7].

Otro tipo de filtro son los filtros digitales los cuales resultan más fáciles de implementar, ya que no requieren de ningún componente físico para su elaboración simplemente se necesita de un proceso computacional o algoritmo en donde una señal digital es transformada en una señal digital nueva [20].

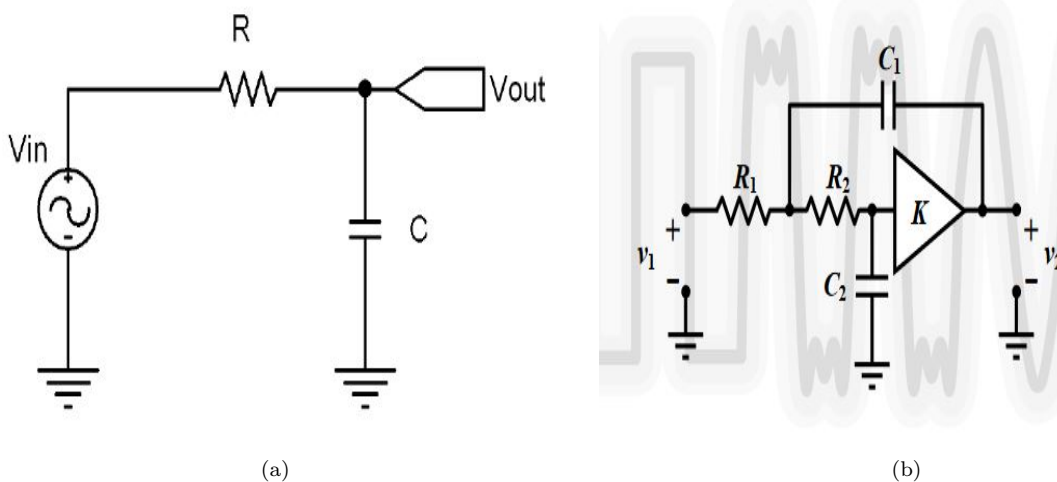


Figura 2.12: configuración circuital de un filtro a) pasivo, b) activo [6][7].

### 2.3.6. Comunicación inalámbrica

La comunicación inalámbrica es aquella en la que ni el emisor ni el receptor se encuentra unidos por un medio físico y se comunican utilizando la modulación de ondas electromagnéticas a través del espacio. En la actualidad las tecnologías inalámbricas tienen una gran aceptación en la industria ya que son sistemas eficientes y de gran facilidad para su manejo [8][21].

El uso de redes inalámbricas son cada vez menos costosos y fáciles de configurar ya que su implementación de hardware y software vienen con una sencilla implementación, se puede destacar las redes que pertenecen al protocolo IEEE 802.15, estas son redes de área personal, en donde está incluida la tecnología bluetooth entre otras. La tecnología bluetooth permite realizar redes Piconet (Red de dispositivos informáticos con un solo maestro), con este tipo de dispositivos se puede transmitir imagen, voz, datos multimedia entre dispositivos portátiles, esta tecnología puede ser utilizada para sistema de control y automatización de procesos industriales [22].



Figura 2.13: Comunicación inalámbrica [8].

### 2.3.7. Definición de los conceptos de la programación estructurada y modular

La programación es el proceso en donde se diseña y se codifica un código fuente con el objetivo de crear un programa que exhiba un comportamiento deseado. Existen varios métodos de programación, los más utilizados son la programación modular y programación estructurada.

- Programación estructurada: Es una técnica de programación la cual consiste en mejorar la claridad, calidad y tiempo de desarrollo de un programa computacional recurriendo a estructuras básicas como secuencias, selección e iteración [23][24].
- Programación modular: La programación modular es un método de programación que consiste en dividir un programa en programas más pequeños o sub programas con el objetivo de resolver tareas tediosas mediante tareas más sencilla permitiendo llegar a una solución general [24].

## Capítulo 3

# DESARROLLO DEL PROYECTO

### 3.1 MÓDULO BLUETOOTH (BT) HC 05

Para la comunicación entre el guante háptico y la pinza robótica se utiliza un sistema inalámbrico el cual consta de dos módulos HC05 configurado el primero como esclavo y el segundo como maestro. Este módulo transmite datos a una velocidad de  $3Mb/s$  con un alcance de quince metros aproximadamente, el cual tiene un voltaje de operación de 3.3V o 5V en la figura 3.1 se muestra el módulo HC05.

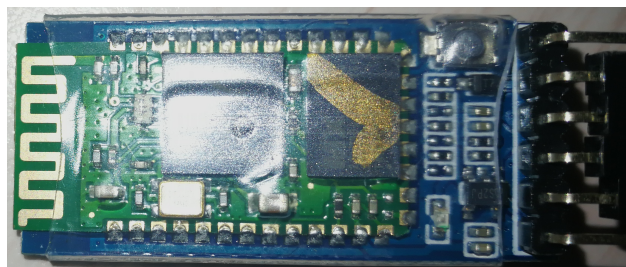


Figura 3.1: Módulo BT HC05

### 3.2 PINZA ROBÓTICA

#### 3.2.1. Montaje de la pinza robótica

Para el montaje del proyecto se utilizó la pinza robótica comercial que se aprecia en la figura 3.2 la cual cuenta con 5 grados de libertad, cada uno de estos tiene un ángulo de giro comprendido entre  $0^\circ$  y  $180^\circ$ , cada grado de libertad está asociado a un servomotor en donde cada servo se definirá como M, donde observando la pinza desde la parte superior hacia la inferior M5 es el cierre y apertura de la pinza, M4 giro de la pinza, M3 movimiento vertical (hacia arriba y abajo), M2 movimiento vertical y M1 movimiento horizontal de toda la pinza (giro de la base), en la cuadro 3.1 se muestran los elementos que componen la pinza.

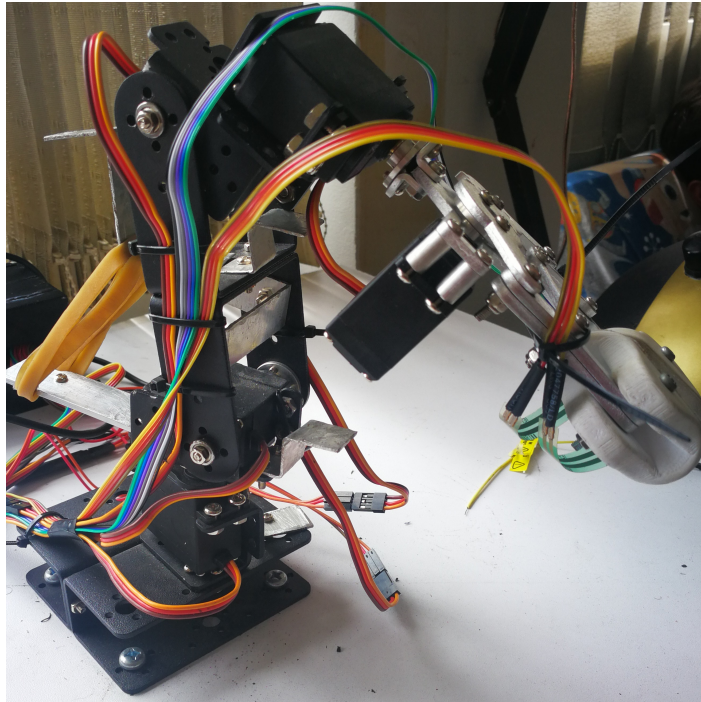


Figura 3.2: Pinza robótica.

Elementos que contiene la pinza robótica	
Soportes multifuncionales	4
Soportes U	3
Soportes U largos	2
Servomotores MG996R	5
Pinza metálica	1
Tornillos y rodamientos	varios

Cuadro 3.1: Elementos de la pinza robótica.

Los servomotores que incluye la pinza robótica están diseñados con una piñonería metálica que les permiten tener un torque de  $9.4\text{Kgf/cm}$  (4.8V) o  $11\text{Kgf/cm}$  (6V), además tienen un sistema de control que los hace muy preciso, contando con tres terminales donde el terminal rojo es VCC, el terminal café o marrón es GND y el terminal naranja es control PWM, por lo general la señal de control para este tipo de servos está en el rango de los 1000ms-2000ms, este puede variar dependiendo del servo y del fabricante, es decir si se genera un pulso de 1000ms equivale a  $0^\circ$  y si se le envía un pulso de 2000ms equivale a  $180^\circ$ , el voltaje de operación de este servo está entre los 4,8V a 7,2V consumiendo una corriente entre 500mA-900mA y una corriente de rotor bloqueado de 2,5A.

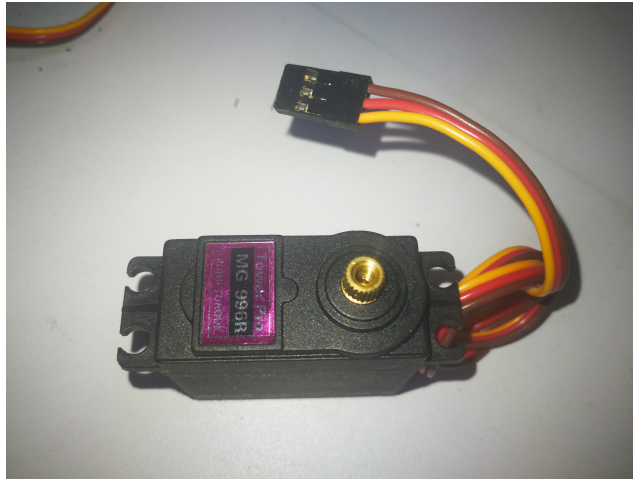


Figura 3.3: Servomotor MG966R.

Este servo tiene un ángulo de giro comprendido entre los  $0^\circ$  y  $180^\circ$  es decir físicamente tienen un tope que no le permite sobrepasar esos ángulos, en caso tal de que estos límites físicos sean sobrepasados el servo puede resultar dañado, por lo cual se optó por implementar algunos topes mecánicos en la pinza robótica como se muestra en la figura 3.4, la función de estos topes es limitar el movimiento de la pinza robótica y de esta forma proteger los servomotores de alguna rotación excesiva y así evitar daños en los servos, también se implementaron bandas elásticas en la parte trasera de la pinza para garantizar el correcto funcionamiento de M2 ya que este no es capaz de retornar a su posición inicial por su propio peso.

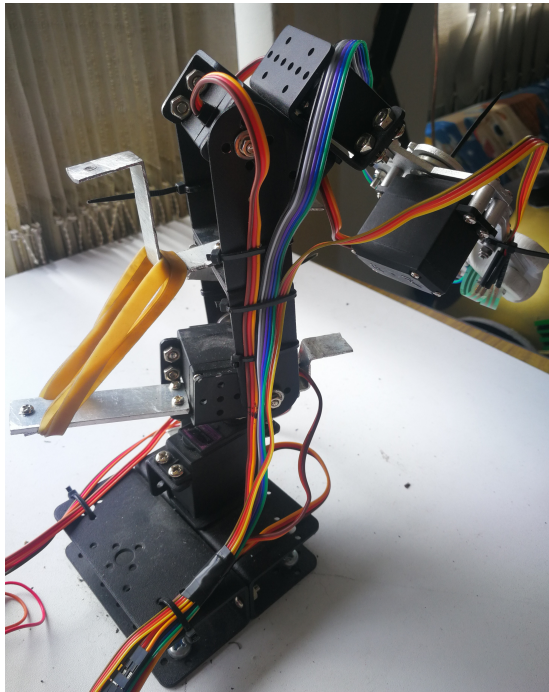


Figura 3.4: Pinza robótica con los topes mecánicos.

### 3.2.2. Sensores en la pinza

Para poder conseguir el efecto háptico es necesario que la pinza robótica cuente con algún tipo de sensor o sensores que le permitan obtener información del objeto que agarre, por lo cual se optó por utilizar dos sensores de fuerza y dos sensores de temperatura. Los sensores de fuerza son los encargados de detectar el contacto de la pinza con algún objeto y los sensores de temperatura se encargan de determinar la temperatura del objeto.

#### 3.2.2.1. Sensor de fuerza (FSR)

El sensor de fuerza que se utiliza para el desarrollo de este proyecto es una resistencia que detecta la fuerza (FSR) de interlink electronics, este es un componente pasivo presentando una disminución de su resistencia al aplicar presión sobre su área sensible la cual tiene un diametro de 0.58" o 1,5cm haciendo lo ideal para simular el tacto en la pinza robótica. Este sensor tienen un rango de operación entre los 2N -20N mostrando una resistencia superior a  $1M\Omega$  cuando no se le aplica presión y una resistencia alrededor de los  $100K\Omega$  cuando se le aplica la mayor cantidad de presión.

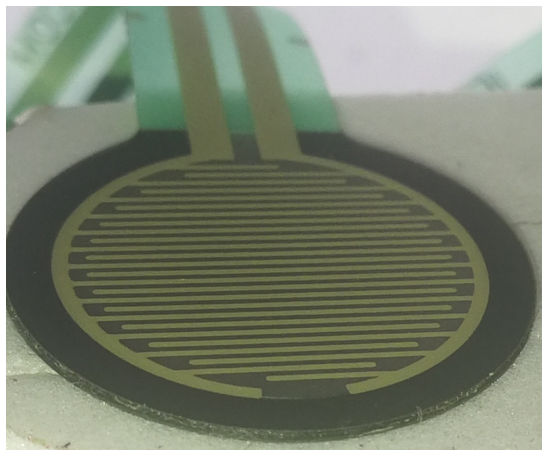


Figura 3.5: Sensor de fuerza FSR 402.

Como se mencionó anteriormente este es un instrumento pasivo (resistivo) por lo cual se debe de implementar un método de instrumentación, el cual permita al arduino leer los datos que entrega el sensor, para esto se utiliza el método de divisor de tensión mencionado en el ítem 2.3.4.

#### 3.2.2.2. Instrumentación del sensor de fuerza (FSR)

Para diseñar el circuito de acondicionamiento de este sensor lo primero es determinar el rango de operación del sensor, este se determina tomando el valor resistivo que entrega el sensor cuando no se le aplica presión sobre el área activa y aplicando la máxima presión sobre el área activa, este mismo procedimiento se realiza para el otro sensor de fuerza ya que son dos sensores, los valores resistivos obtenidos se muestran en la siguiente tabla.

Valores resistivos(KΩ)		
	sin aplicar presión	aplicando presión
Sensor 1	1000	0,4
Sensor 2	1000	0,42

Cuadro 3.2: Valores resistivos máximos y mínimos de los sensores de fuerza.

Ya determinado el rango de operación de los sensores se procede a determinar la resistencia promedio ( $R_{serie}$ ), está es la que va en serie con el sensor, la cual se calcula de la siguiente manera.

$$R_{max} = R_{sinaplicarpresión} \quad (3.1)$$

$$R_{min} = R_{aplicarpresión} \quad (3.2)$$

$$R_{serie} = \frac{R_{max} + R_{min}}{2} \quad (3.3)$$

De esta manera se obtiene el valor resistivo de la resistencia serie para cada uno de los sensores, por ejemplo, si se desea calcular el valor de la resistencia serie del sensor 1 se procede de la siguiente forma.

$$R_{max} = 1M\Omega \quad (3.4)$$

$$R_{min} = 400\Omega \quad (3.5)$$

$$R_{serie} = \frac{1M\Omega + 400\Omega}{2} = 500,2K\Omega \quad (3.6)$$

El siguiente paso luego de haber encontrado los valores de las resistencias serie para los dos sensores es determinar el rango de operación en voltios de cada sensor.

Rango de voltaje (V)		
	sin aplicar presión	aplicando presión
Sensor 1	0	3,57
Sensor 2	0	3,45

Cuadro 3.3: Valores máximos y mínimos de voltajes entregados por los sensores.

Obtenidos los rangos máximos y mínimos de voltaje que nos arroja el acondicionamiento de señal se procede a adquirir los datos por medios del conversor análogo-digital del arduino, este puerto tiene un rango



de operación de 0v a 5v permitiendo digitalizar los valores arrojados por el acondicionamiento. La función de arduino que permite adquirir los valores arrojados por el acondicionamiento se muestra en la figura 3.6.

```
void setup() {
    // put your setup code here, to run once:

}

void loop() {
    // put your main code here, to run repeatedly:
    int fuerzal;
    fuerzal= analogRead(A0); //lectura del puerto analogo digita cero
}
```

Figura 3.6: Código para la lectura del puerto análogo-digital del arduino.

Luego de realizar la lectura con el conversor análogo digital del arduino se obtienen los valores digitalizados que se muestran en el cuadro 3.4.

Valores digitales		
	sin aplicar presión	aplicando presión
Sensor 1	0	731
Sensor 2	0	706

Cuadro 3.4: Valores digitales.

Estos valores digitales son enviados vía BT al guante háptico el cual cuenta con dos pequeños motores DC uno ubicado en el pulgar y el otro en el dedo medio permitiendo al operador tener la realimentación vibratoria del sistema, es decir, si a uno de los sensores o a los dos se le aplica presión en el área activa, el operador por medio de pequeñas vibraciones se percata de dicha acción que se está realizando en la pinza.

### 3.2.3. Sensor de temperatura LM35

Para detectar la temperatura del objeto que agarra la pinza se hace uso del sensor de temperatura LM35, el cual permite medir temperaturas de  $-55^{\circ}\text{C}$  a  $150^{\circ}\text{C}$  con un factor de escala lineal de  $10\text{mV}/^{\circ}\text{C}$ . Para procesar los datos entregado por el sensor se hace uso del puerto análogo-digital del arduino nuevamente, el cual cuenta con una resolución de 4.9 milivoltios, permitiendo detectar los cambios de temperatura que entrega el sensor. La pinza tendrá incorporados dos de estos sensores para garantizar una mejor lectura de la temperatura del objeto que agarra. Los valores digitalizados serán enviados al guante háptico que es el encargado de transmitir la sensación térmica al operador.





Figura 3.7: Sensor de temperatura LM35.

Los rangos de operación de los sensores que se van a manejar en este proyecto se muestran en el siguiente cuadro.

Rango de operación (°C)		
	mínimo	máximo
Sensor 1	0	150
Sensor 2	0	150

Cuadro 3.5: Rangos de operación de los sensores expresados en °C.

Los rangos mostrados en el cuadro anterior ahora son expresados en voltios como se muestra en el siguiente cuadro.

Rango de operación (V)		
	mínimo	máximo
Sensor 1	0	1,5
Sensor 2	0	1,5

Cuadro 3.6: Rangos de operación de los sensores expresados en voltios (V).

Como se menciona anteriormente la señal entregada por los sensores debe ser procesada por el puerto análogo-digital del arduino y enviados al guante háptico, en el siguiente cuadro se observan los valor del cuadro 3.6 pero digitalizados.

Rango de operación digital		
	mínimo	máximo
Sensor 1	0	307
Sensor 2	0	307

Cuadro 3.7: Rangos de operación de los sensores digitalizados.

### 3.2.4. Adaptación de los sensores a la pinza robótica

Para la adaptación de los sensores de fuerza y temperatura se optó por diseñar una pieza que permita adaptar estos sensores en la pinza, en la figura 3.8 y en la figura 3.9 se muestran las piezas que se diseñaron.

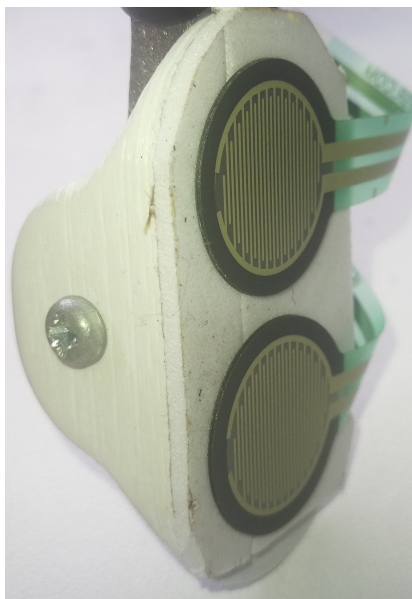


Figura 3.8: Adaptador para la pinza con los sensores de fuerza.

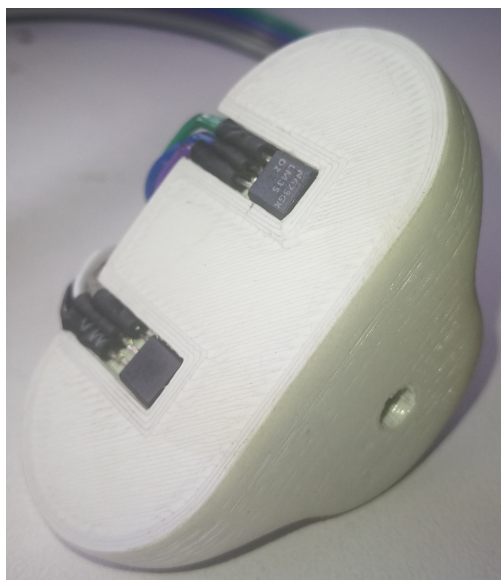


Figura 3.9: Adaptador para la pinza con los sensores de temperatura.

Estas adaptaciones que se le realizan a la pinza permiten obtener información de temperatura y fuerza cuando está agarrando un objeto, en la Figura 3.10 se muestra la pinza con las piezas montadas.

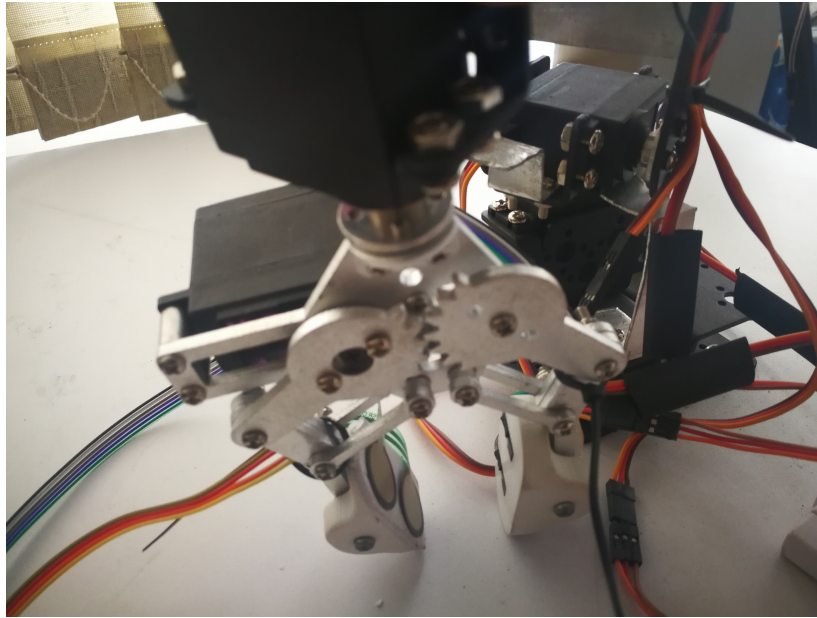


Figura 3.10: Pinza robótica con las piezas y los sensores montados.

### 3.2.5. Sistema de control de la pinza robótica

El Sistema que rige o controla la pinza robótica es un arduino nano, este es el encargado por medio del módulo BT de recibir los datos de posición para cada uno de los servos o grados de libertad enviados por el guante háptico y a su vez enviar los valores sensados de temperatura y fuerza hacia el guante háptico. La pinza cuenta con 5 grados de libertad por lo cual se debe de generar una señal de control (PWM) para cada grado de libertad, es decir, el guante háptico enviara la posición (ángulo) para cada uno de los servos y por medio de software se le da la orden al servo que se mueva a dicho ángulo, la función de arduino que permite realizar dicha acción se muestra en la figura 3.11.

```
void setup() {
  m1.attach(pin_m1,1000,2000); //tiempo para m1
}

void loop() {
  m1.write(angulo); // angulo del servo
  delayMicroseconds(2000);
}
```

Figura 3.11: Función de control para el servo.

La acción mostrada en la figura anterior se debe de repetir para los cinco servos que cuenta la pinza robótica, en donde cada uno de ellos será controlado desde el guante háptico.

### 3.2.6. Circuito Electrónico para el control de la pinza robótica

En la figura 3.12 se muestra el esquemático del circuito implementado para el control de la pinza robótica y en la figura 3.13 se observa la board del circuito. Se decidió tener dos alimentaciones para este diseño, en donde una alimentación es para la etapa de potencia (los servo motores) y otra para la etapa de control (el arduino).

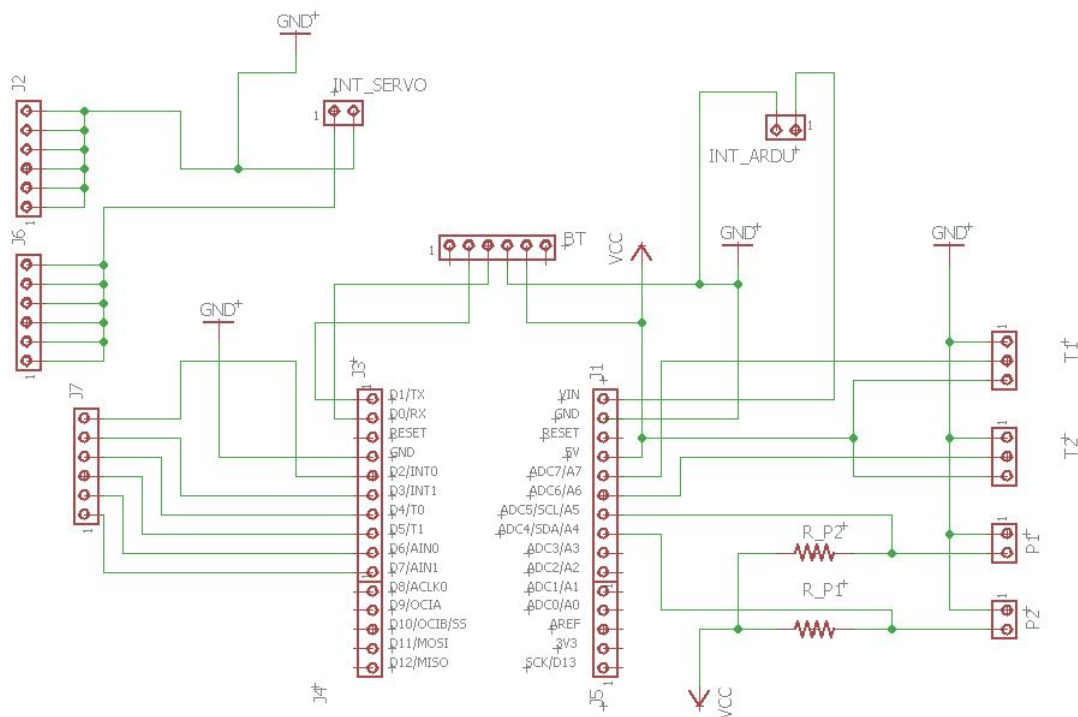


Figura 3.12: Esquemático del circuito de control de la pinza robótica.

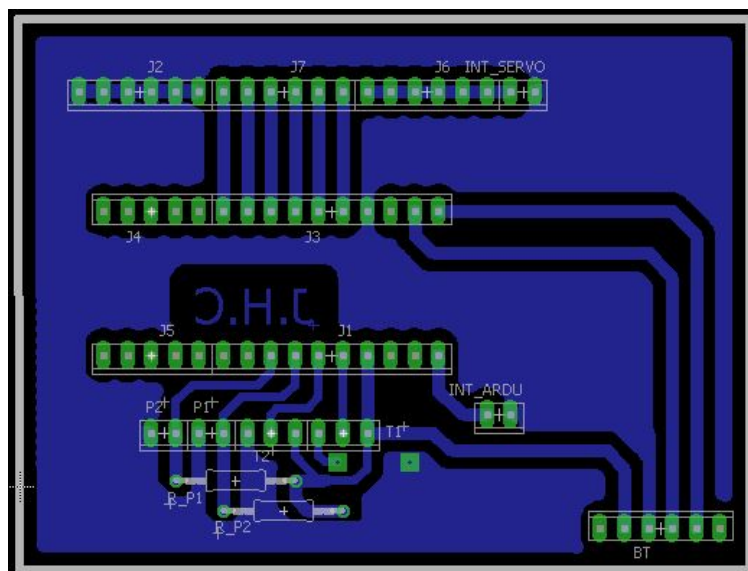


Figura 3.13: Board del circuito de control de la pinza robótica.

En la siguiente figura se observa la pcb con los elementos ya montados, también se diseñó una caja en la cual va la circuitería de la pinza robótica.

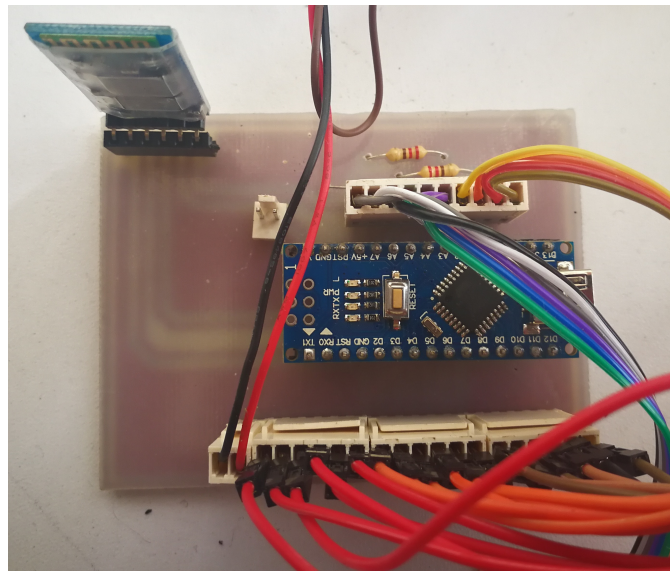


Figura 3.14: PCB del circuito de control de la pinza robótica.



Figura 3.15: Caja para la circuitería de la pinza.

A continuación se muestra el montaje final de todo la pinza.



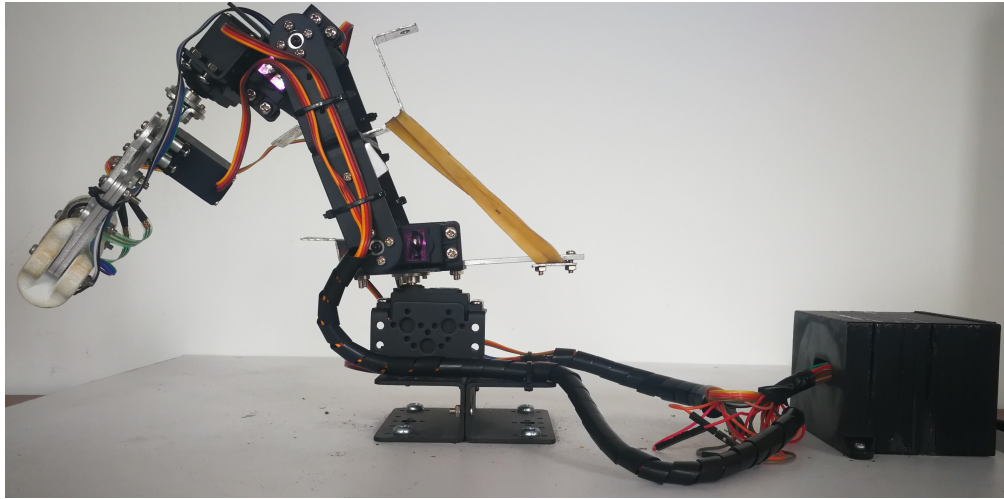


Figura 3.16: Montaje completo de la pinza robótica

### 3.3 GUANTE HÁPTICO

Este guante háptico tiene incorporados dos pequeños motores DC uno en el dedo pulgar y el otro en el dedo medio, los cuales generan pequeñas vibraciones permitiendo tener el efecto háptico en el guante, este motor tiene un voltaje máximo de operación de 3V y cuenta con un tamaño de 8mm de diámetro haciéndolo ideal para este tipo de aplicaciones, el guante también cuenta con una resistencia calefactora ubicada en la parte superior de la mano, la cual le permite al operador tener la sensación térmica, esta resistencia maneja un voltaje operación de 12V máximo y tiene un diámetro de 5cm. Estas dos sensaciones vibratorias y caloríficas son las componentes hápticas que trae incorporada el guante. En la figura 3.17 se muestra el motor DC utilizado y en la figura 3.18 la resistencia calefactora.



Figura 3.17: Motor DC incorporado en el guante háptico.



Figura 3.18: Resistencia calefactora incorporado en el guante háptico.

### 3.3.1. Sistema de control en el guante háptico

Para que la pinza robótica pueda imitar los movimientos realizado por el operador se implementa un sistema de control basado en un guante como se muestra en la figura 3.19, donde este está constituido por dos módulos MPU 6050 los cuales permiten sensar los movimientos realizados por la muñeca y el brazo, también cuenta con dos resistencias flexibles una encargada de controlar el agarre de la pinza y la otra la rotación de toda la pinza, también incluye dos motores DC, una resistencia calefactora de los cuales ya se habló en el item anterior y un arduino nano que es el encargado de procesar todas la señales y trasmitirlas hacia la pinza, este guante tiene la capacidad de controlar los 5 servos o grados de libertad con los que cuenta la pinza.



Figura 3.19: Guante háptico implementado en el proyecto.

### 3.3.2. Arduino en el guante háptico

El sistema encargado de toda la parte de control en el guante háptico es un arduino nano ver figura 3.20, este tiene la tarea de realizar todas las lecturas de los sensores, hacer el debido procesamiento de señales, filtrar, enviar y recibir todos los datos vía BT.

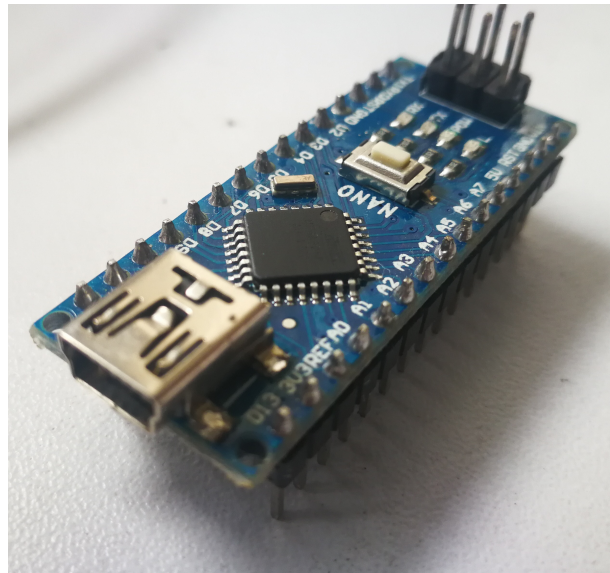


Figura 3.20: Arduino nano V3.0.

En el cuadro 3.8 se muestra las características del arduino nano.



<b>Especificaciones del arduino nano</b>	
<b>Microcontrolador</b>	ATmega328
<b>Voltaje Operativo</b>	5V
<b>Voltaje de Entrada</b>	7-12V
<b>Voltaje de Entrada(límites)</b>	6-20V
<b>Pines digitales de Entrada/Salida</b>	14 (de los cuales 6 proveen salida PWM)
<b>Pines análogos de entrada</b>	8
<b>Corriente DC por cada Pin Entrada/Salida</b>	40 mA
<b>Corriente DC entregada en el Pin 3.3V</b>	50 mA
<b>Memoria Flash</b>	36 KB (2KB usados por el bootloader)
<b>SRAM</b>	2KB
<b>EEPROM</b>	4KB
<b>Clock Speed</b>	16 MHz
<b>Dimisiones</b>	0,73"x1,70"

Cuadro 3.8: Especificaciones del arduino nano V3.0.

### 3.3.3. Control del motor DC y la resistencia calefactora en el guante

Para poder lograr el efecto háptico como se mencionó anteriormente el guante cuenta con dos pequeños motores DC y una resistencia calefactora.

#### 3.3.3.1. Motor DC

Para poder controlar la intensidad de vibración del motor se hace uso de la salida PWM del arduino, está salida tiene un rango de operación de 0V-5V, la función encargada de realizar esta operación se muestra en la figura 3.21. Esta función maneja dos parámetros uno es el puerto por el cual se va a enviar la señal PWM y el otro parámetro es el valor de salida que se desea escribir, el cual maneja un rango de 0 a 255 donde 0 equivale a 0V y 255 a 5V, en el cuadro 3.4 se muestran los valores que son enviados desde la pinza hacia el guante háptico (motor DC) los cuales deben de ser escalados, ya que la función **analogwrite** tiene como parámetro máximo de salida 255, por este motivo los valores mostrados en el cuadro 3.4 son divididos entre 4, esto permite que los valores sensados estén dentro del rango de 0 a 255, en el cuadro 3.9 se muestran los nuevos rangos que serán escritos o enviados vía BT hacia el guante.

Un factor muy importante que hay que tener en cuenta es que este motor DC maneja un voltaje máximo de operación de 3V, por lo cual por medio de software se limita la salida del puerto análogo tal que no supere este valor.

```

void setup() {
    // put your setup code here, to run once:

}

void loop() {
    // put your main code here, to run repeatedly:
    analogWrite(9,sensor1/4);
    |
}

```

Figura 3.21: Función analogwrite de arduino.

	Valores digitales	
	sin aplicar presión	aplicando presión
Sensor 1	0	182
Sensor 2	0	176

Cuadro 3.9: Valores escalados para los motores DC.

### 3.3.3.2. Resistencia calefactora

Antes de generar el efecto térmico hay que tener en cuenta algunos aspectos muy importantes. Los seres humanos tenemos una tolerancia máxima y mínima de temperatura, este es un aspecto que se debe de tener en cuenta para no causar alguna lesión al operador, ya que el objetivo es tener una realimentación térmica pero sin generar ningún tipo de quemadura al operador. El ser humano puede percatarse si un objeto está caliente cuando dicho objeto supera la temperatura corporal la cual está alrededor de los 37°C.

Si se habla de temperatura máxima y mínima externa que puede soportar el cuerpo humano, se puede decir que la temperatura máxima que llega a soportar el cuerpo humano está entre los 45°C a los 50°C, una temperatura mayor puede ocasionar daños colaterales y la temperatura mínima que puede soportar el cuerpo humano es de -10°C, en esta ocasión solo nos interesa la temperatura máxima, ya que la resistencia calefactora no puede exceder ese rango de temperatura.

Otro factor que se debe tener en cuenta de la resistencia es que al ser térmica su respuesta es lenta, es decir que esta se demora cierto tiempo en alcanzar la temperatura deseada, por lo cual se optó por mantenerla caliente alrededor de los 34°C y de esta forma obtener una respuesta mas rápida cuando la pinza agarre algún objeto caliente.

Se manejará una temperatura máxima para la resistencia calefactora alrededor de los 45°C, con el objetivo de no causar ninguna lesión al operador.

### 3.3.4. Lectura de la resistencia flexible

El guante háptico tendrá incorporado dos flex sensor o resistencias flexibles, una ubicada en el pulgar y la otra en el dedo medio, el sensor que está ubicada en el pulgar se encarga de controlar el giro de la pinza robótica (M1) y el otro del agarre (M5).

Como se mencionó anteriormente en el ítem 2.3.3 la resistencia flexible es un elemento pasivo el cual entregar un valor resistivo, estos valores no pueden ser leídos directamente por el arduino, por ende se debe

de implementar algún método de acondicionamiento de señal que le permitan al arduino leer los valores entregados por el sensor. El acondicionamiento que se utiliza para este sensor es el método de divisor de tensión.

El primer paso a seguir es determinar el rango de operación de los sensores, los cuales se determinan realizando los movimiento de extensión y contracción para los dos sensores.

Valores resistivos(K $\Omega$ )		
	Extensión	contracción
Corazón	32.3	72.3
pulgar	36.8	62.3

Cuadro 3.10: . Rangos de operación de los sensores expresado en ohmios ( $\Omega$ ).

Luego de obtener los valores máximos y mínimos de los sensores se procede a determinar el valor de la resistencia serie con el objetivo de realizar el acondicionamiento de los sensores, a continuación se hace el cálculo para la resistencia que está ubicada en el dedo medio, este mismo procedimiento se debe de realizar para el otro sensor.

$$R_{max} = R_{contracción} \quad (3.7)$$

$$R_{min} = R_{extensión} \quad (3.8)$$

$$R_{serie} = \frac{R_{max} + R_{min}}{2} = \frac{72,3K\Omega + 32,3K\Omega}{2} = 52,3K\Omega \quad (3.9)$$

Luego de obtener el valor promedio se realiza el acondicionamiento y se determina el rango de operación en voltios que entrega este acondicionamiento.

Medición de voltaje (v)		
	Dedo estirado	Dedo flexionado
Corazón	1,9	2,5
pulgar	2.28	2.93

Cuadro 3.11: Rangos de operación de los sensores expresado en voltios (V).

Luego de hallar el rango de operación en voltios del acondicionamiento se procese a digitalizar los valores de igual manera que se hizo con el sensor de fuerza para ser enviados a la pinza robótica.

Valores digitales		
	Dedo estirado	Dedo flexionado
Corazón	388	511
pulgar	467	600

Cuadro 3.12: Rangos de operación de los sensores digitalizados.

Estos dos valores mostrados en el cuadro anterior son los valores digitales entregados por el acondicionamiento del sensor flexible, como se mencionó anteriormente estos sensores serán los encargados del giro y el agarre de la pinza robótica, estos deben de ser enviados a la pinza pero antes de realizar esta acción hay que realizar un escalamiento, ya que los valores mostrados en el cuadro 3.12 deben de estar comprendidos entre  $0^{\circ}$ - $180^{\circ}$ , para realizar el escalamiento de estos valores se hace uso de la función `map()` de arduino, esta función recibe tres parámetros, el primero es el dato con el que se va a trabajar el que se desea escalar, el segundo es el rango de valores en el que se va a mover el primer parámetro y el tercero el rango de salida que se quiere obtener, permitiendo obtener un fácil escalamiento de la señal ver figura 3.22.

```
datos[5]=map(|datos[5], 388, 511, 20, 70|);
```

Figura 3.22: Código de la función `map()` de arduino.

### 3.3.5. Uso del acelerómetro dentro del guante háptico

La pinza robótica tiene 5 grados de libertad donde M1 y M5 son controlados con las resistencias flexibles ubicadas en el guante háptico quedando 3 grados de libertad restantes, M4 y M3 son controlados por un acelerómetro ubicado en la parte superior de guante (en la muñeca), M2 por el otro acelerómetro ubicado en el bicep del operador.

#### 3.3.5.1. Implementación del acelerómetro de la IMU MPU 6050

Para medir los movimientos realizado por el operador se emplea el módulo IMU MPU 6050 el cual se mencionó en el ítem 2.3.2 este dispositivo tiene seis grados de libertad permitiendo realizar múltiples tareas, con un voltaje de operación  $2.375V - 3.46V$  el cual posee una interface que le permite operar hasta los  $5V$ , haciéndolo ideal para muchas tarjetas de desarrollo que operan en este rango, este módulo maneja el estándar de comunicación I2C facilitando su conexión.

El MPU 6050 posee un giroscopio y un acelerómetro los cuales pueden realizar medidas en los tres ejes. El giroscopio permite realizar mediciones de velocidad angular con diferentes sensibilidades de  $\pm 250^{\circ}/s$ ,  $\pm 500^{\circ}/s$ ,  $\pm 1000^{\circ}/s$  y  $\pm 2000^{\circ}/s$ , el acelerómetro permite medir las aceleraciones en los tres ejes y de igual forma que el giroscopio se puede ajustar la sensibilidad entre  $\pm 2g$ ,  $\pm 4g$ ,  $\pm 8g$  y  $\pm 16g$ , internamente posee un conversor analógico-digital de 16bits permitiendo una alta precisión en la conversión de los datos entregados por los sensores. Para una fácil implementación de este módulo en el proyecto, se diseñó una pieza para hacer el montaje del acelerómetro en el guante como se muestra en la figura 3.23.

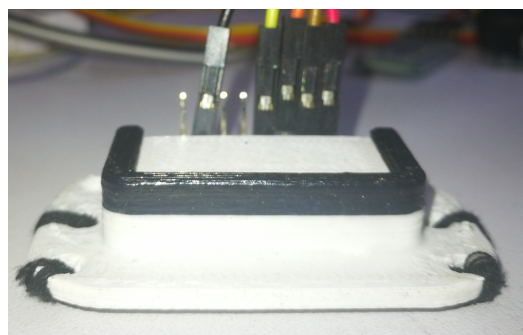


Figura 3.23: Módulo MPU 6050 dentro de la pieza.

### 3.3.5.2. Medición de los ángulos de inclinación

Con el acelerómetro ubicados en la muñeca y el otro en el antebrazo se puede medir los movimientos realizados por estas extremidades permitiendo controlar el resto de la pinza robótica, la IMU ubicada en la muñeca controlara M4 y M3, la que está ubicada en el antebrazo controlará M2 .

Si el módulo MPU 6050 se encuentra en la posición que se muestra en la figura 3.24 considerando esta como la posición de reposo y teniendo en cuenta que sobre el actúa la fuerza de la gravedad se obtiene un vector con el valor de  $1g$  en la componente Z y de  $0g$  en las componentes X y Y, si se le realiza un giro de  $90^\circ$  al sensor, se obtiene una fuerza gravitatoria de  $1g$  en la componente X y  $0g$  en las otras dos componentes ver figura 3.25 y si se vuelve a rotar  $90^\circ$  se obtiene una componente de  $1g$  en el eje Y y  $0g$  en las otras dos componentes [25].

Estas proyecciones que se presenta al rotar el modulo son las que permiten determinar el ángulo de inclinación presenta en cada una de las componentes. Para encontrar esta inclinación se hace uso de los ángulos directores, estos permiten determinar la dirección y el sentido de un vector [26][27].

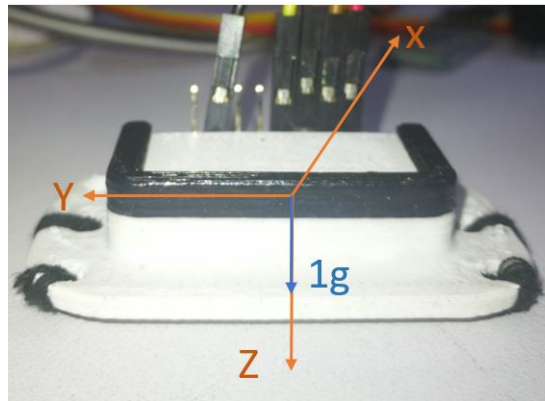


Figura 3.24: Posición de reposo de la IMU MPU 6050 con su componente gravitatoria en Z.

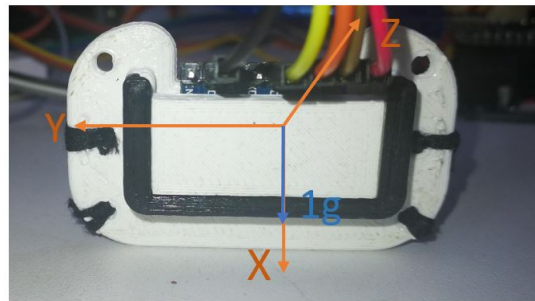


Figura 3.25: IMU MPU 6059 rotada  $90^\circ$  con su componente gravitatoria en X.

### 3.3.5.3. Medición del ángulo de inclinación para los servos M4 y M3

La ubicación de la IMU MPU 6050 en el guante permite medir los movimientos realizados por esta extremidad, como se muestra en la figura 3.26, siendo esta la posición de reposo se obtienen las siguientes componentes gravitatoria de la IMU.

$$X1 = 0g$$

$$Y1 = 0g$$

$$Z1 = 1g$$

Con estas proyecciones se puede hallar la posición de la IMU o la posición relativa de la muñeca del operador, esto se determina de la siguiente manera.

$$\Theta_{x1} = \arccos \frac{0g}{1g} = 90^\circ$$

$$\Theta_{y1} = \arccos \frac{0g}{1g} = 90^\circ$$

$$\Theta_{z1} = \arccos \frac{1g}{1g} = 0^\circ$$

Teniendo en cuenta que los movimientos que nos interesa registrar son los del eje X y Y el eje Z no se utilizará, para generar los ángulos reales con los cuales se van a operar se le deben de restar  $90^\circ$  a  $\Theta_{x1}$  y  $\Theta_{y1}$  es decir que los ángulos  $\Theta_{x1}$  y  $\Theta_{y1}$  marcaran el valor de  $0^\circ$  en su posición de reposo, estos ángulos tendrán un rango de operación entre  $-90^\circ$  a  $90^\circ$ .

$$\Theta_{x1} = \arccos \frac{0g}{1g} = 0^\circ$$

$$\Theta_{y1} = \arccos \frac{0g}{1g} = 0^\circ$$

$$\Theta_{z1} = \arccos \frac{1g}{1g} = 0^\circ$$

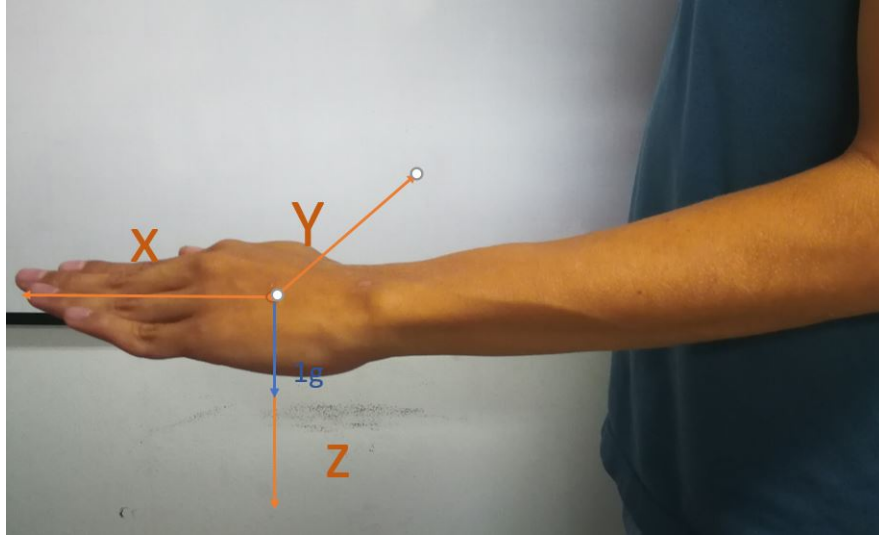


Figura 3.26: Posición de reposo del operador con las componentes gravitatorias en la muñeca.

Si se realiza una inclinación en el Eje X de  $-45^\circ$  ver figura 3.27 en donde se generan las siguientes proyecciones.

$$X1 = 0,707g$$

$$Y1 = 0g$$

$$Z1 = 1g$$

Los ángulos directores obtenidos son los siguiente.

$$\Theta_{x1} = \arccos \frac{0,707g}{1g} = 45^\circ$$

$$\Theta_{y1} = \arccos \frac{0g}{1g} = 90^\circ$$

$$\Theta_{z1} = \arccos \frac{0,707g}{1g} = 45^\circ$$

Restando le los  $90^\circ$  a los ángulos  $\Theta_{x1}$  y  $\Theta_{y1}$  nos queda la posición real del acelerómetro.

$$\Theta_{x1} = \arccos \frac{0,707g}{1g} = -45^\circ$$

$$\Theta_{y1} = \arccos \frac{0g}{1g} = 0^\circ$$

$$\Theta_{z1} = \arccos \frac{0g}{1g} = 45^\circ$$

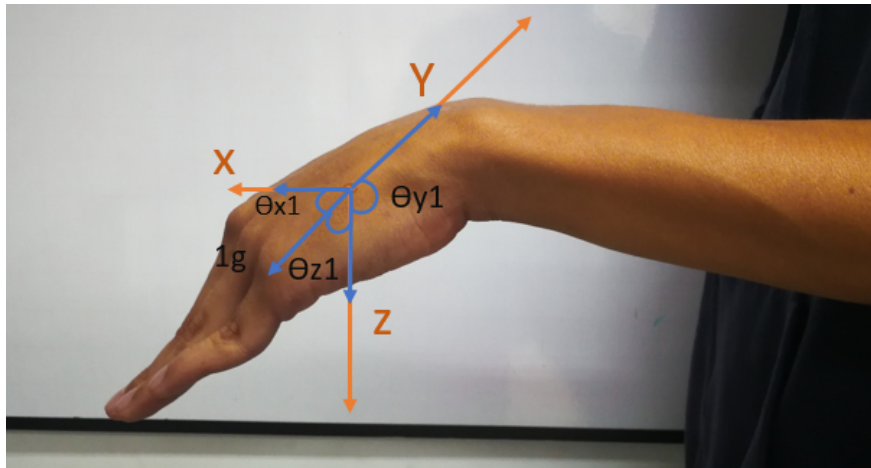


Figura 3.27: Posición a  $45^\circ$  de la muñeca.

Ahora si el operador realiza una inclinación en el eje X y también en el eje Y se obtendrán las siguientes proyecciones como se muestra en la figura 3.28.

$$\Theta_{x1} = \arccos \frac{-0,707g}{1g} = 135^\circ$$

$$\Theta_{y1} = \arccos \frac{-0,707g}{1g} = 135^\circ$$

$$\Theta_{z1} = \arccos \frac{-0,707g}{1g} = 135^\circ$$

Restando los  $90^\circ$  a los ángulos  $\Theta_{x1}$  y  $\Theta_{y1}$  nos queda.

$$\Theta_{x1} = \arccos \frac{-0,707g}{1g} = 45^\circ$$

$$\Theta_{y1} = \arccos \frac{-0,707g}{1} = 45^\circ$$

$$\Theta_{z1} = \arccos \frac{0,707g}{1g} = 135^\circ$$

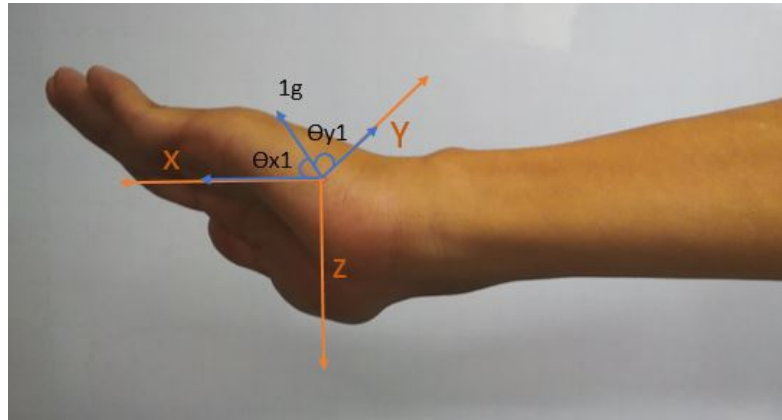


Figura 3.28: Inclinación a  $45^\circ$  de la muñeca en los ejes X y Y.

#### 3.3.5.4. Medición del ángulo de inclinación para el servo M2

El acelerómetro ubicado en el bicep es el encargado de registrar los movimientos realizados por este musculo el cual controlara el servo M2, teniendo en cuenta que solo se registraran los movimientos efectuados en el Eje X no se tendrán en cuenta los otros dos ejes, de igual forma que se procedió en el ítem anterior para hallar los ángulos directores se procederá en este ítem.

Siendo la posición de reposo la que muestra en la figura 3.29 se obtienen las siguientes proyecciones.



$$X_2 = 1g$$

$$Y_2 = 0g$$

$$Z_2 = 0g$$

Los ángulos directores se calculan de la siguiente manera.

$$\Theta_{x2} = \arccos \frac{1g}{1g} = 0^\circ$$

$$\Theta_{y2} = \arccos \frac{0g}{1g} = 90^\circ$$

$$\Theta_{z2} = \arccos \frac{0g}{1g} = 90^\circ$$

Siendo la posición de reposo la que se muestra en la figura 3.29 y con los ángulos directores obtenidos, no es necesario restar ningún valor a los ángulos.



Figura 3.29: Posición de reposo del operador con las componentes gravitatorias en el bicep.

Si el operador realiza el movimiento que se observa en la figura 3.30 se obtendrán las siguientes proyecciones.

$$X_2 = 0g$$

$$Y_2 = 0g$$

$$Z_2 = 1g$$

Los ángulos directores se calculan de la siguiente forma.

$$\Theta_{x2} = \arccos \frac{0g}{1g} = 90^\circ$$

$$\Theta_{y2} = \arccos \frac{0g}{1g} = 90^\circ$$

$$\Theta_{z2} = \arccos \frac{1g}{1g} = 0^\circ$$

Y de esta forma se obtiene el angulo de desplazamiento en el eje X de la extremidad.

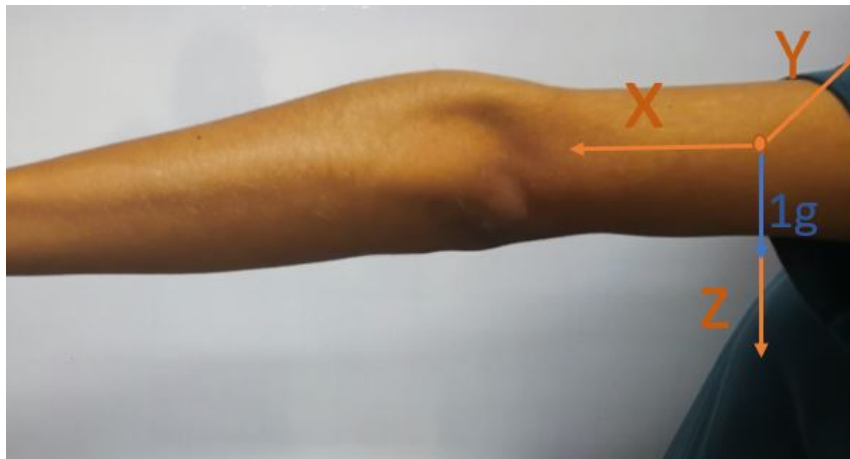


Figura 3.30: Movimiento del bicep a  $90^\circ$ .

### 3.3.6. Circuito Electrónico para el control del guante háptico

Para el guante háptico se construyó una sola tarjeta electrónica la cual estará ubicada en la cintura del operador, en esta tarjeta está ubicado el arduino nano y el módulo BT, para la comunicación de la placa con los sensores se utilizó una cinta de cables la cual permite comunicarse con todos los sensores sin la necesidad de que estén en la misma placa. En la Figura 3.31 se muestra el esquemático del circuito de control y en la figura 3.32 se muestra el board. Para la alimentación del circuito y la resistencia calefactora se utiliza una batería recargable de Li-Po de 11.1V a 2.2A.



En la figura 3.33 se muestra la PCB del circuito con sus elementos, para este circuito también se diseñó una caja la cual dentro de ella va el circuito y la batería como se muestra en la Figura 3.34.

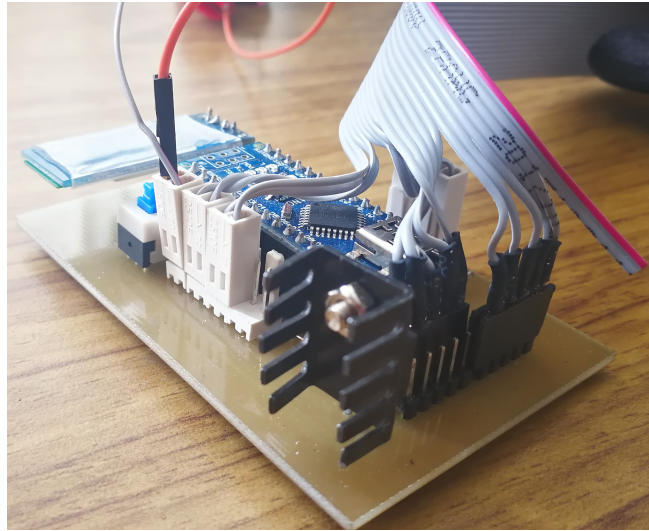


Figura 3.33: Montaje completo del circuito de control del guante.



(a)



(b)

Figura 3.34: a) caja con la PCB, b) montaje de la caja con la PCB y la batería.

En la siguiente figura se muestra el operador utilizando todo el sistema de control del guante.



Figura 3.35: Sistema de control del guante completado.



## Capítulo 4

# PRUEBAS Y RESULTADOS

### 4.1 PRUEBAS DE FUNCIONAMIENTO DE LOS SERVO MOTORES

Para comprobar el correcto funcionamiento del guante háptico se realizaron las siguientes pruebas.

Como se ha mencionado en el texto el operador puede controlar remotamente la pinza, la cual cuenta con 5 grados de libertad donde cada uno de ellos tiene un rango de operación ya establecido, los objetos que puede manipular tienen un peso igual o inferior a los 100g según las pruebas realizadas, además la pinza no podrá sujetar objetos que tengan un grosor superior a los 3.3cm. Inicialmente se comienza con la posición de reposo como se puede observar en la figura 4.1.



Figura 4.1: Posición inicial del sistema.

Luego el operador realiza un movimiento vertical hacia arriba y abajo de la muñeca (M3) como se observa en las siguientes figuras y de forma inmediata la pinza realiza los movimientos echos por el operador en donde a M3 se le estableció un rango de operación de  $10^{\circ}$  a  $155^{\circ}$ .



Figura 4.2: Prueba del movimiento vertical de la muñeca a  $-45^{\circ}$ (M3).



Figura 4.3: Prueba del movimiento vertical de la muñeca a  $90^{\circ}$ (M3).



En la figura 4.4 se observa el giro de la muñeca (M4) donde la pinza sigue el movimiento realizado por el operador, a M4 se le estableció un rango de operación de  $20^{\circ}$  a  $160^{\circ}$ .

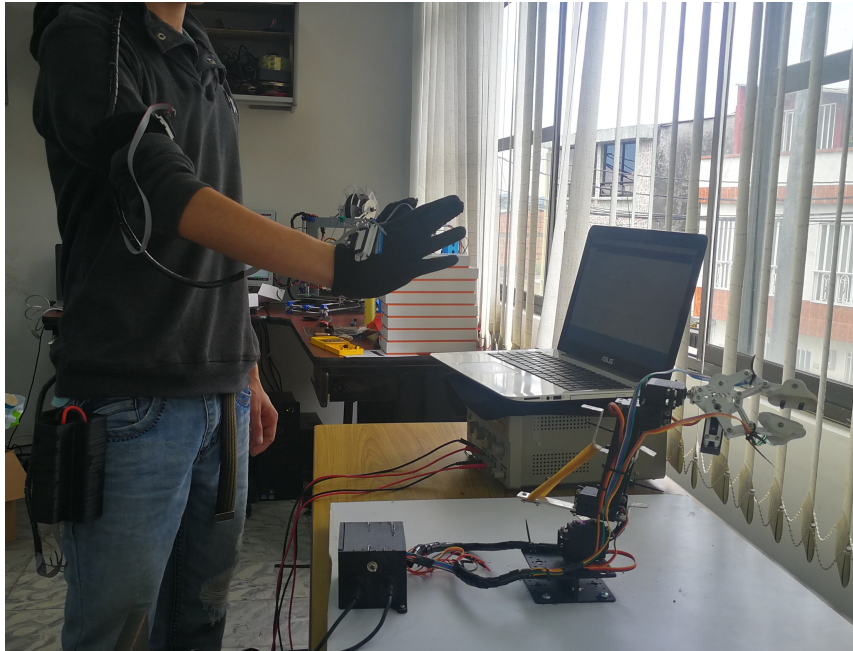


Figura 4.4: Prueba del movimiento rotatorio de la muñeca en sentido horario(M4).

La IMU ubicada en el bicep del operador se encarga de controlar el servo M2, cuando el operador estira o contrae el brazo se realiza un movimiento vertical (M2) como se muestra en la Figura 4.5, el rango de operación para M2 es de  $80^{\circ}$  a  $150^{\circ}$ .

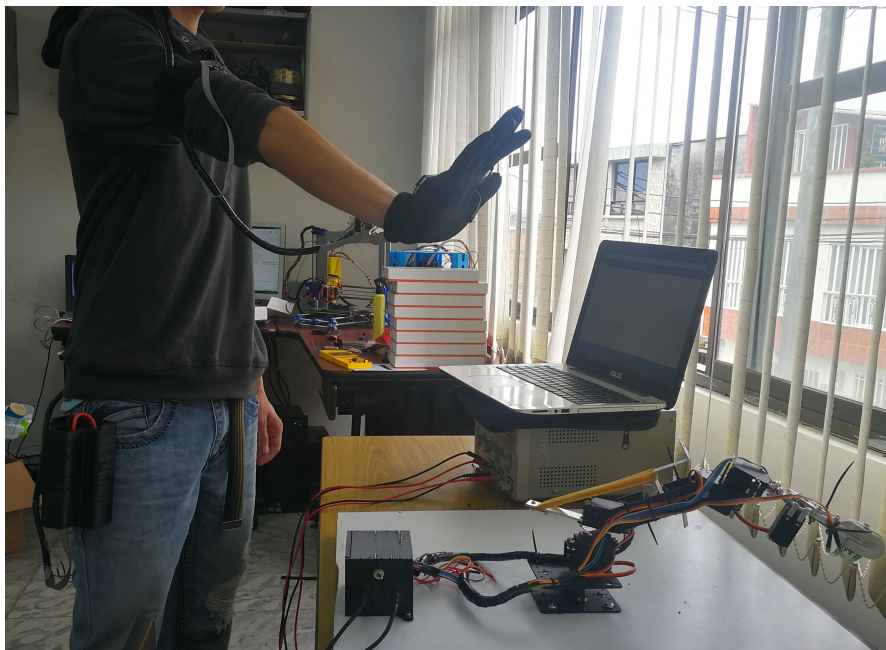


Figura 4.5: Prueba del movimiento vertical del bicep (M2).



Con el dedo pulgar el operador puede controlar el movimiento rotatorio (M1) de la pinza con se muestra en las siguientes figuras, con un rango de operación de  $15^\circ$  a  $165^\circ$ ..

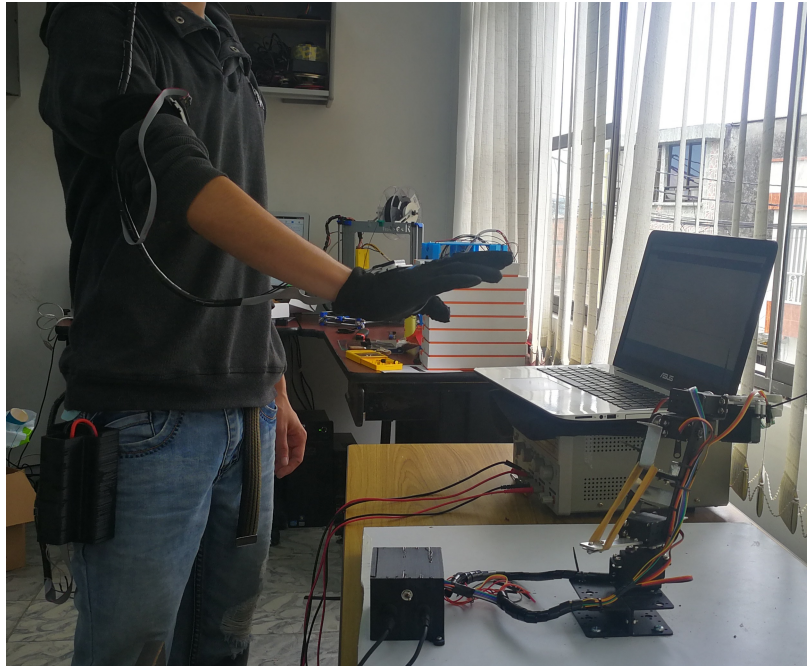


Figura 4.6: Prueba del movimiento rotatorio de la base en el sentido antihorario (M1).

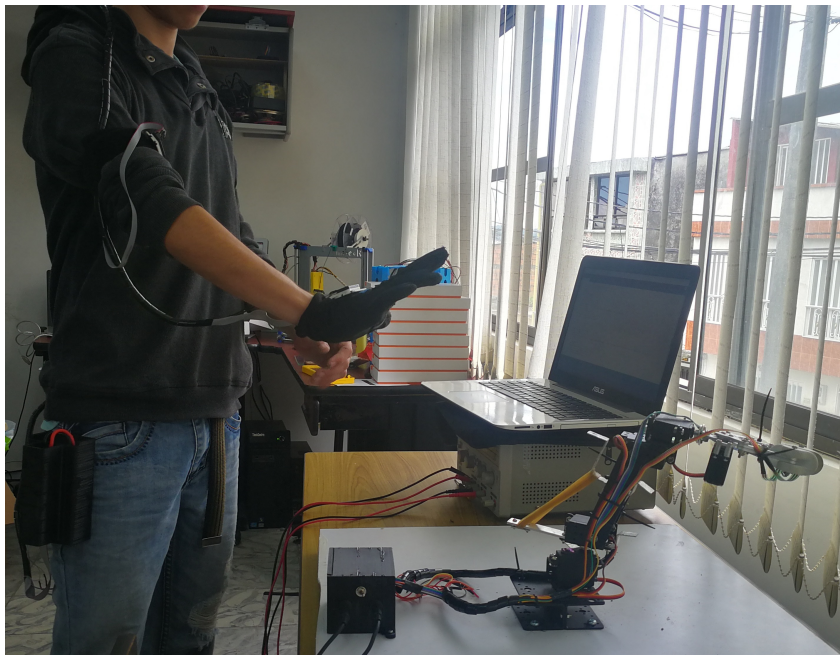


Figura 4.7: Prueba del movimiento rotatorio de la base en el sentido horario (M1).

## 4.2 PRUEBAS HÁPTICAS DEL SISTEMA

A continuación, se muestran los resultados de las pruebas hápticas del sistema en donde el operador toma algún objeto y obtiene una realimentación vibratoria en el guante, como se mencionó en el ítem 3.2.3. La pinza cuenta con dos sensores de temperatura para determinar la temperatura a la que se encuentran los objetos que se están manipulando, este sensor puede soportar una temperatura máxima de  $150^{\circ}\text{C}$ , una situación que debe de tenerse en cuenta a la hora de manipular el sensor de fuerza esta relacionado con el hecho de que para trabajar adecuadamente sobre dicho sensor no debe actuar una temperatura superior a los  $85^{\circ}\text{C}$  lo que define la cuota máxima de trabajo del sensor de fuerza. Adicionalmente se sabe que la temperatura corporal de un ser humano adulto varia entre los  $36.1^{\circ}\text{C}$  y los  $37.2^{\circ}\text{C}$ , por lo tanto, para que el operador pueda tener una realimentación térmica en el guante la pinza debe sujetar un objeto cuya temperatura sea igual o superior a los  $37^{\circ}\text{C}$  e inferior a los  $85^{\circ}\text{C}$ , donde el guante se calentará alrededor de los  $45^{\circ}\text{C}$ .

### 4.2.1. Pruebas vibratorias

Para realizar las pruebas vibratorias se utilizó un prototipo de medición de fuerza desarrollado en el grupo de investigación robótica aplicada ver figura 4.8, el cual consiste en una celda de carga a la cual se le aplica una fuerza, cuyo resultado se muestra a través de una pantalla LCD esta es medida en Newton , de esta forma se puede determinar la fuerza que puede aplicar la pinza a los objetos que sujeta.

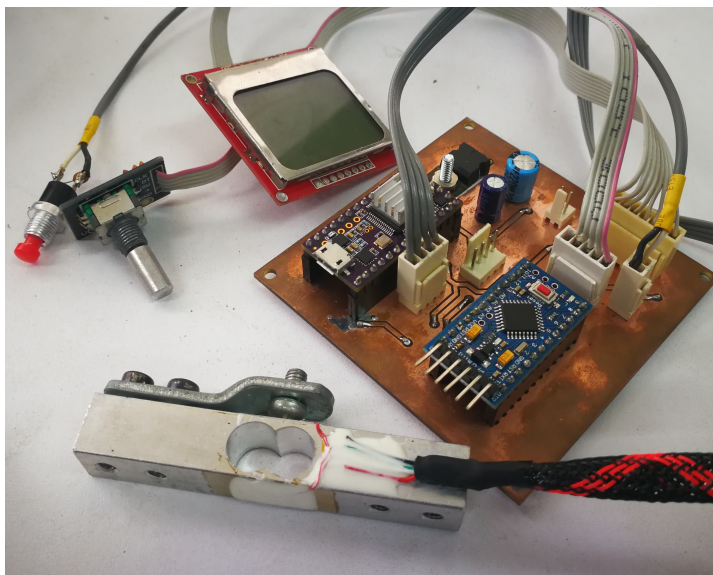
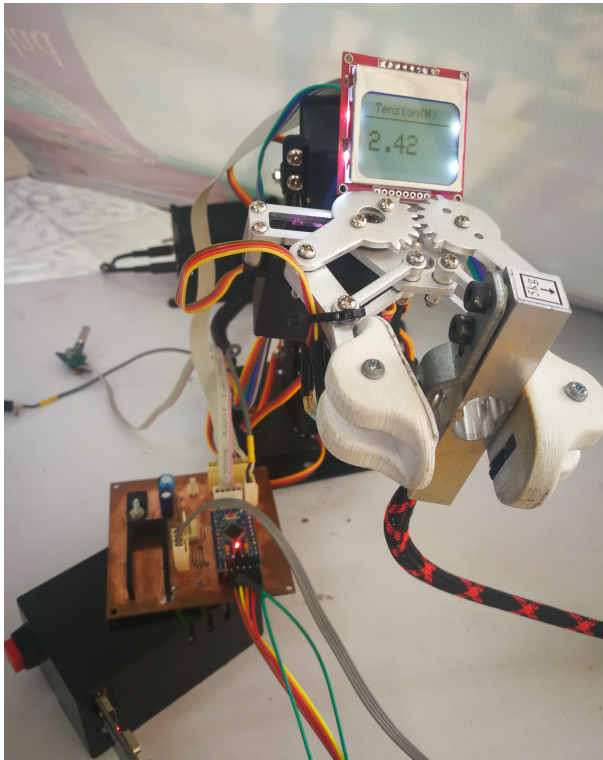
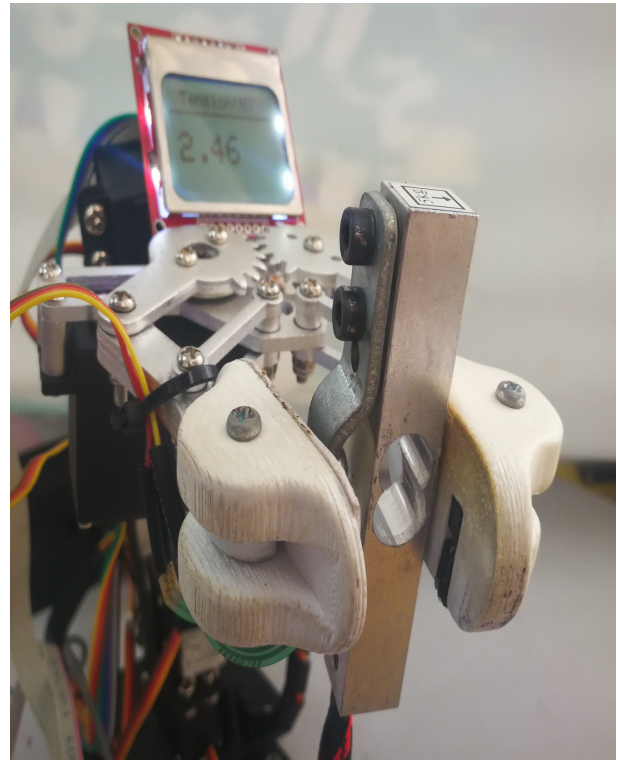


Figura 4.8: Prototipo de medición de fuerza.

Al sistema que se observa en la figura 4.8, se le aplicaron diferentes fuerzas con la pinza robótica con el objetivo de establecer el comportamiento de los sensores de fuerza y saber cual es la fuerza máxima que puede aplicar la pinza a los objetos que sujeta, la cual es de  $6\text{N}$ , en la figura 4.9 se observan las pruebas realizadas con el sistema.



(a)



(b)

Figura 4.9: Pruebas de fuerza realizadas con el prototipo de medición de fuerza.

Entre mayor fuerza aplique la pinza a los objetos que sujeta mayor sera el voltaje, este voltaje se ve reflejado en la percepción vibratoria en el guante, es decir, cuando se presenta una variación de voltaje esto implica que en la salida PWM varíe su ciclo de trabajo (Duty), en donde estas variaciones de voltaje son inyectadas a los motores DC ubicados en guante, permitiendo tener diferentes percepciones vibratorias. En la cuadro 4.1 se observan las diferentes fuerzas aplicadas, el equivalente de esas fuerzas en vol y el ciclo de trabajo que representa ese voltaje.

Fuerza (N)	Voltaje(V)	Duty (%)
1,28	0,31	6,20
1,55	0,35	7,00
2,15	0,40	8,00
2,95	0,51	10,20
4,79	0,63	12,60
4,90	0,68	13,60

Cuadro 4.1: Relación entre fuerza, voltaje y Duty.



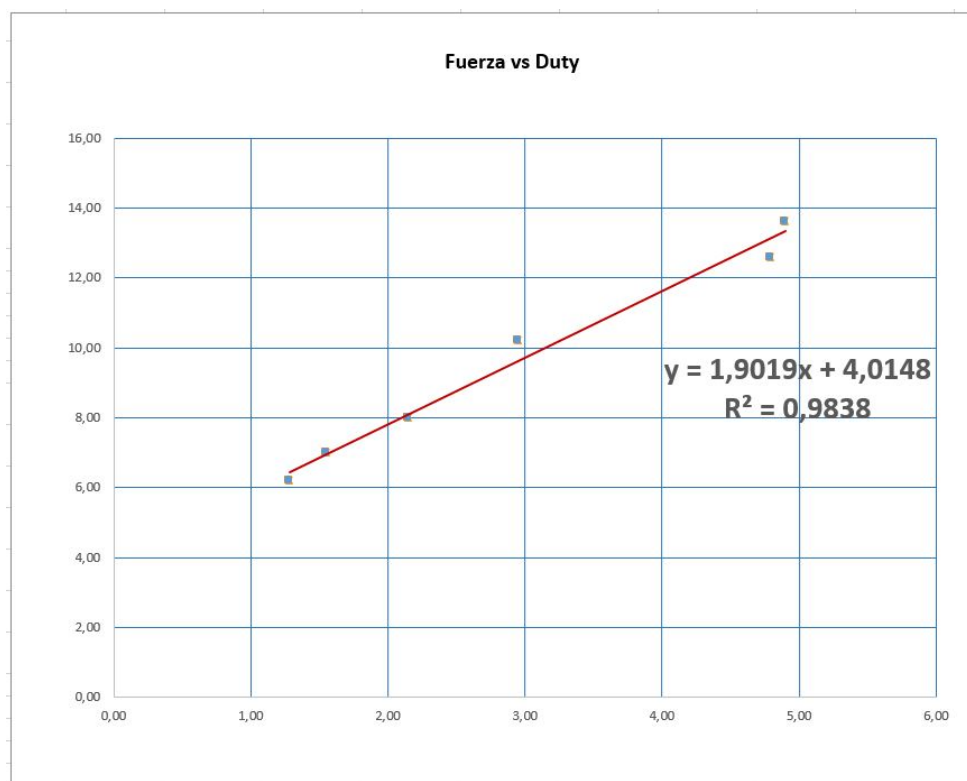


Figura 4.10: Grafica de Fuerza vs Duty.

En la gráfica 4.10 se observa la relación lineal entre la fuerza aplicada por la pinza robótica vs el ciclo de trabajo, donde el eje Y representa el ciclo de trabajo (%) y el eje X la fuerza aplicada en Newton (N).

Como se observa en el cuadro 4.1 los voltajes obtenidos son pequeños por lo cual se optó por darle un offset de 1,5v como se observa en el cuadro 4.2, lo cual permite conserva la relación entre fuerza y voltaje, y a su vez aumentar la percepción vibratoria en el guante.

Fuerza (N)	Voltaje(V)	Duty (%)
1,28	1,81	36,20
1,55	1,85	37,00
2,15	1,90	38,00
2,95	2,01	40,20
4,79	2,13	42,60
4,90	2,18	43,60

Cuadro 4.2: Relación entre fuerza, voltaje y Duty con offset incluido.

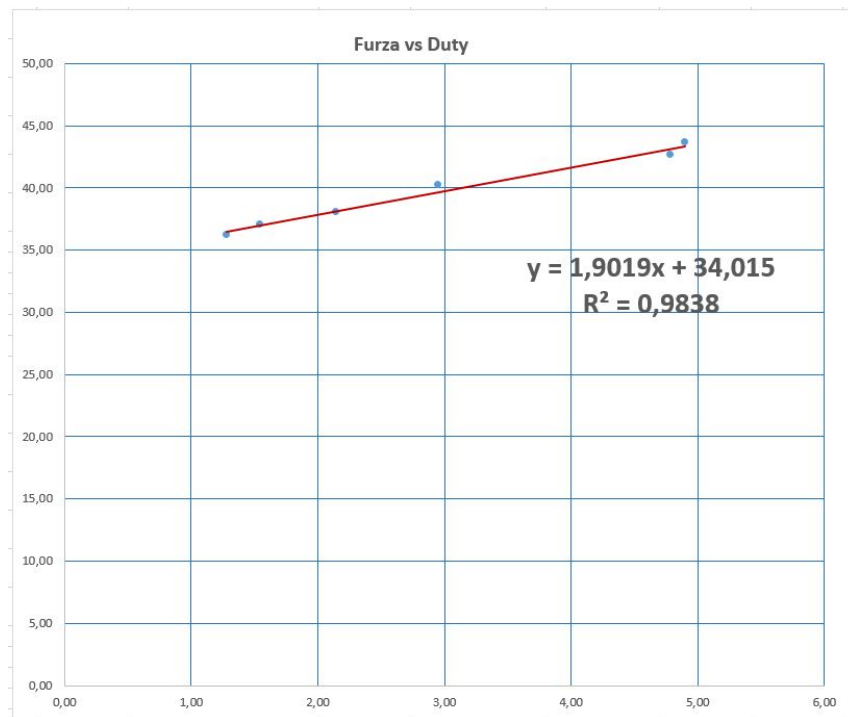


Figura 4.11: Grafica de Fuerza vs Duty con offset.

En la figura 4.12 se observa al operador manipulando diferentes objetos.



(a)



(b)

Figura 4.12: Pruebas vibratorias del sistema hápticas.

#### 4.2.2. Pruebas térmicas

Como se mencionó en el ítem 3.3.3.2 cuando la pinza no está sensando ningún objeto que supere la temperatura de los 37°C la resistencia calefactora se mantiene alrededor de los 34°C con se muestra en la figura 4.13 con el objetivo de tener una respuesta térmica más rápida cuando la pinza agarre algún objeto caliente.

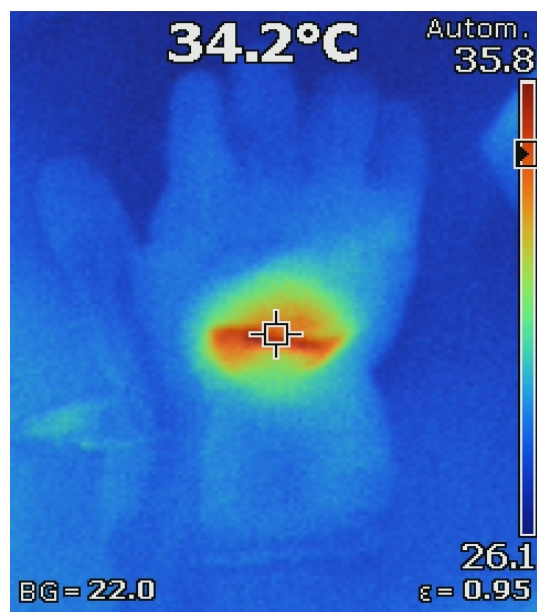


Figura 4.13: Temperatura presente en el guante cuando no se detecta ningún objeto caliente.

Cuando la pinza sensa un objeto con temperatura superior a los 37°C la resistencia calefactora se empieza a calentar alrededor de los 45°C como se muestra en la figura 4.14.

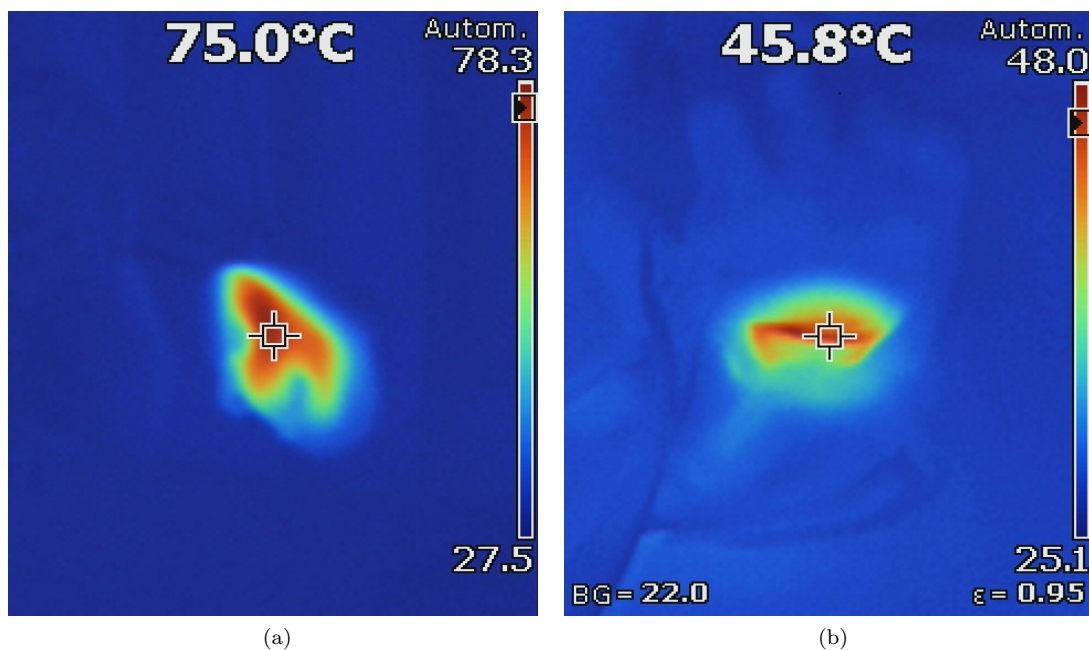


Figura 4.14: a) temperatura presente en la pinza, b) temperatura presente en el guante.

## Capítulo 5

# CONCLUSIONES Y FUTUROS TRABAJOS

### 5.1 CONCLUSIONES

- la estructura mecánica de la pinza robótica no cuenta por si misma con la fuerza suficiente para garantizar todos los movimientos que se requieren, esta situación puede generar incluso daños en el sistema mecánico, por tal razón se implementaron bandas elásticas unidas a la estructura que le aportan fuerza extra para garantizar su funcionamiento.
- Seleccionar adecuadamente la tecnología y lenguaje de programación con la cual se va a desarrollar el proyecto es de suma importancia ya que esto permite reducir tiempo y la cantidad de líneas de códigos, por esta razón se escogió la plataforma de desarrollo arduino la cual incluye una gran cantidad de librerías lo que permite una mayor facilidad a la hora de implementar el código.
- Debido a que el rango de percepción de temperatura de una persona sin que sufra daño alguno es pequeño ( $45^{\circ}\text{C}$  -  $50^{\circ}\text{C}$ ) se optó por calentar la resistencia térmica a una temperatura alrededor de los  $45^{\circ}\text{C}$  cuando la pinza sujeta algún objeto con temperatura entre los  $37^{\circ}\text{C}$  y  $85^{\circ}\text{C}$ , sin establecer una relación directa entre la temperatura que sensa la pinza y la temperatura que se le transmite al operador.
- Los resultados obtenidos en las pruebas demuestran que el sistema háptico cumple con los objetivos propuestos; esto debido a que se verificó su adecuado funcionamiento, lo anterior se puede observar en la validación del sistema en la relación fuerza mecánica en la pinza y vibración en el guante, temperatura en la pinza y generación de temperatura en el guante.
- El módulo bluetooth HC-05 tiene un alcance de 15m, sin embargo, en las pruebas de control remoto realizadas se encontró que para el sistema construido opera adecuadamente hasta un valor máximo de 10m, de ese valor en adelante se empieza a generar una pérdida en la transmisión de datos.

## 5.2 FUTUROS TRABAJOS

- Implementar cámaras en el sistema háptico desarrollado a fin de establecer una mejor relación entre el operador y la pinza robótica.
- Cambiar la tecnología bluetooth implementada en este diseño por otra, como por ejemplo radiofrecuencia, lo cual permitiría que el sistema pueda ser controlado a una mayor distancia.
- Adaptar la pinza a una plataforma móvil con el objetivo que pueda moverse en un entorno mas amplio, dando al sistema mayor grado de libertad.
- Implementar algún tipo de sistema de realimentación que permita mejorar el control de los sistemas desarrollados.
- Equipar a la pinza con más y mejores sensores a fin de elevar su desempeño.



## Capítulo 6

# PRESUPUESTO

El presupuesto utilizado para este proyecto se muestra en el siguiente cuadro.

Rubro	cantidad	Valor unitario (COP)	valor total
arduino nano	2	\$ 20.000	\$ 40.000
tarjeta MPU 6050	2	\$ 10.000	\$ 20.000
resistencia flexible	2	\$ 37.000	\$ 74.000
resistencia calefactora	1	\$ 56.970	\$ 56.970
motor vibrador para celular	2	\$ 1.493	\$ 2.986
sensor LM35	2	\$ 2.350	\$ 4.700
modulo bluetooth	2	\$ 16.000	\$ 32.000
kit de resistencias	1	\$ 11.280	\$ 11.280
regleta de pines	2	\$ 1.000	\$ 2.000
pinza robótica	1	\$ 350.000	\$ 350.000
baquelita de fibra de vidrio	1	\$ 10.000	\$ 10.000
termo encogible	1	\$ 2.000	\$ 2.000
sensor de fuerza	2	\$ 21.000	\$ 42.000
batería Li-Po 3s	1	\$ 100.000	\$ 100.000
ABS	1	\$ 100.000	\$ 100.000
cable ribbon	1	\$ 9.000	\$ 9.000
total			\$ 856.936

Cuadro 6.1: presupuesto.

# Bibliografía

- [1] J. W. Genoy-Muñoz, L. F. Rodríguez-Ortiz, S. A. Salinas *et al.*, “Interfaz háptica de cuatro grados de libertad para aplicaciones quirúrgicas,” 2014. (document), 2.1, 2.1
- [2] K. v. z. m. Sandra angelica pusil arce, “Posicionamiento de lapbot en un ambiente tridimensional virtual usando interfaz h  ptica,” 2010. (document), 2.1, 2.2
- [3] M. A. Nieto, E. E. Navia Z  niga, G. A. Araque, R. Prieto Ladino, G. Botero, N. Octavio, M. E. Rojas, G. Valderrama, J. Jhair, A. F. Loaiza Carre  o *et al.*, “Memorias del simposio de investigaci  n e innovaci  n y desarrollo tecnol  gico, del centro industrial y desarrollo empresarial de soacha,” 2017. (document), 2.1, 2.3
- [4] C. Glove, “Cyberglove systems inc,” 2016. (document), 2.1, 2.4, 2.1
- [5] “<https://www.dji.com/phantom-3-pro>,” <https://www.dji.com/phantom-3-pro>, Apr. 2015, 20-07-2018. (document), 2.6
- [6] F. Miyara, “Filtros activos,” *E-book, Rosario: Universidad Nacional de rosario*, 2004. (document), 2.3.5, 2.12
- [7] A. Autor and I. Autor, “T  tulo del art  culo,” *Nombre de la*, 2003. (document), 2.3.5, 2.12
- [8] J. Andreu, *Redes inal  mbricas (Servicios en red)*. Editex, 2011. (document), 2.3.6, 2.13
- [9] J. Mart  nez, J. P. Molina, A. S. Garc  a, D. Mart  nez, and P. Gonz  lez, “Desarrollo de un guante de datos con retorno h  ptico vibro-t  ctil basado en arduino,” *Jornadas de Realidad Virtual*, pp. 1–10, 2009. 1.1
- [10] R. L. Hollis, “Butterfly haptics: A high-tech startup [entrepreneur],” *IEEE Robotics Automation Magazine*, vol. 17, no. 4, pp. 14–17, Dec. 2010. 1.1
- [11] M. R. D  az-Tribaldos, J. M. Escobar-Ocampo, and   . A. Vivas-Alb  n, “Interfaz h  ptica tipo guante con realimentaci  n vibratoria,” *Revista EIA*, vol. 12, no. 23, pp. 29–39, 2015. 1.1
- [12] F. Rothling, R. Haschke, J. J. Steil, and H. Ritter, “Platform portable anthropomorphic grasping with the bielefeld 20-dof shadow and 9-dof tum hand,” in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 2951–2956. 1.1
- [13] “Angus-rob  tica aplicada,” <https://sites.google.com/site/gruporoboticaaplicada/home/proyectos/PROYECTOS-ROBTICA/ANGUS>, Jul. 2015, 10-07-2018. 2.1
- [14] “transductores,” <http://www.ehu.eus/acustica/espanol/electricidad/transes/transes.html>, 2003, 20-07-2018. 2.3.1
- [15] A. Serna, F. Ros, and J. Rico, *Gu  a pr  ctica de sensores*. Creaciones Copyright SL, 2010. 2.3.1
- [16] D. L  pez Montes *et al.*, “Fusi  n sensorial para la estimaci  n del estado de un v  h  culo aut  nomo,” 2016. 2.3.2, 2.3.2, 2.3.2.1, 2.3.2.2
- [17] M. RIVAS, “Unidad de medida inercial: Algoritmo de estimaci  n e implementaci  n software,” *Proyecto Final de Carrera. Universidad de Sevilla. Escuela T  cnica Superior de Ingenieros. Departamento de Ingenier  a Electr  nica. Espa  a*, 2006. 2.3.2

- [18] M. A. Arenas, J. M. Palomares, L. Girard, J. Olivares, and J. M. Castillo-Secilla, “Diseño y construcción de un guante de datos mediante sensores de flexibilidad y acelerómetro,” 2011. 2.3.3
- [19] J. Sánchez, “Circuitos de acondicionamiento para sensores resistivos,” 2011. 2.3.4
- [20] E. G. Gutiérrez, “Introducción al filtrado digital,” *Catalunya: Escola Superior de Musica de Catalunya, Departamento de Sonologia*, 2009. 2.3.5
- [21] E. Martínez, “La evolución de la telefonía móvil,” *Revista Red*, vol. 1, pp. 1–6, 2001. 2.3.6
- [22] C. DURÁN ACEVEDO and R. A. CASTRO MIRANDA, “Comunicación inalámbrica basada en tecnología bluetooth para la automatización de procesos industriales,” *El Hombre y la Máquina*, no. 39, 2012. 2.3.6
- [23] H. M. Deitel and P. J. Deitel, *Como programar en Java*. Pearson Educación, 2003. 2.3.7
- [24] J. B. Carpio, *Curso práctico de programación en C y C++*. Universitat Jaume I, 2001, no. 8. 2.3.7
- [25] A. I. Bhuyan and T. C. Mallick, “Gyro-accelerometer based control of a robotic arm using avr microcontroller,” in *Strategic Technology (IFOST), 2014 9th International Forum on*. IEEE, 2014, pp. 409–413. 3.3.5.2
- [26] F. P. Beer, E. R. Johnston, E. R. Eisenberg, and W. E. Clausen, *Mecánica vectorial para ingenieros*. McGraw-hill, 1990, no. 968-422-565-2. 04-A1 LU. CG-12. 3.3.5.2
- [27] A. Engler, D. Müller, and M. Hecklein, “Geometría analítica,” *Ediciones UNL. Santa Fé, Argentina*, 2005. 3.3.5.2

# Apéndice A

## CÓDIGOS DESARROLLADOS

El código desarrollado para el sistema de control del guante y de la pinza se implementaron en la plataforma arduino.

### A.1 CÓDIGO IMPLEMENTADO PARA EL GUANTE

```
1 #include <SPI.h>
2 #include <Wire.h>
3 #include "I2Cdev.h"
4
5 const int MPU_1 = 0x68; // I2C dirección de la MPU-6050
6 const int MPU_2 = 0x69; // I2C dirección de la MPU-6050
7 const double resolucion = 0.000061035;
8 const double resol_giro = 0.007633587;
9
10 //variables que entregan las MPU 6050 de aceleración
11 int16_t AcX, AcY, AcZ; //aceleración
12 int16_t AcX2, AcY2, AcZ2;
13
14 //variable para recibir y enviar los datos del BT
15 String R = ""; //variable en donde se van a guardar los datos recibidos por el BT
16 float datos[5]; // variable donde se van a guardar los datos recibidos por el BT
17
18 //variables para el filtro
19 double angulo1_X_fil = 0.0, angulo1_Y_fil = 0.0, angulo1_Z_fil = 0.0;
20 double angulo1_X2_fil = 0.0, angulo1_Y2_fil = 0.0, angulo1_Z2_fil = 0.0;
21
22
23
24 void MPU_1_acele() {
25     Wire.beginTransmission(MPU_1);
26     Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
27     Wire.endTransmission(false);
28     Wire.requestFrom(MPU_1, 6, true); //A partir del 0x3B, se piden 6 registros
29     AcX = Wire.read() << 8 | Wire.read(); //Cada valor ocupa 2 registros
30     AcY = Wire.read() << 8 | Wire.read();
31     AcZ = Wire.read() << 8 | Wire.read();
32 }
```

```

33 void MPU_2_acele() {
34     Wire.beginTransaction(MPU_2);
35     Wire.write(0x3B); //Pedir el registro 0x3B - corresponde al AcX
36     Wire.endTransmission(false);
37     Wire.requestFrom(MPU_2, 6, true); //A partir del 0x3B, se piden 6 registros
38     AcX2 = Wire.read() << 8 | Wire.read(); //Cada valor ocupa 2 registros
39     AcY2 = Wire.read() << 8 | Wire.read();
40     AcZ2 = Wire.read() << 8 | Wire.read();
41 }
42
43 // Funciones para calcular los cosenos directores -----
44 double Ax( double x, double y, double z ) {
45     double h = sqrt ( x * x + y * y + z * z );
46     double angulo = (180 / PI) * acos ( x / h ) - 90.0;
47
48     /*if( angulo >= 180.0) {
49         angulo = 180.0;
50     } else if( angulo <= 0.0 ){
51         angulo = 0.0;
52     */
53
54     return angulo ;
55 }
56
57 double Ay( double x, double y, double z ) {
58     double h = sqrt ( x * x + y * y + z * z );
59     double angulo = (180 / PI) * acos ( y / h ) - 90.0 ;
60
61     /*if( angulo >= 180.0) {
62         angulo = 180.0;
63     } else if( angulo <= 0.0 ){
64         angulo = 0.0;
65     */
66
67     return angulo ;
68 }
69
70 double Az( double x, double y, double z ) {
71     double h = sqrt ( x * x + y * y + z * z );
72     double angulo = (180 / PI) * acos ( z / h ) ;
73
74     /*if( angulo >= 180.0) {
75         angulo = 180.0;
76     } else if( angulo <= 0.0 ){
77         angulo = 0.0;
78     */
79     return angulo;
80 }
81
82 // Filtro pasa bajos
83 double filtro_pasa_bajos(double entrada_actual, double salida_anterior ,
84
85     const double Wc = 5.0;
86     const double To = 50.0 / 1000.0;

```

```

87     const double a = Wc / ( (1 / To) + Wc );
88
89     return ( a * entrada_actual + (1 - a) * salida_anterior );
90 }
91 // Fin del filtro pasa bajos
92
93
94 void setup() {
95     // put your setup code here, to run once:
96     Serial.begin(115200);
97
98     Wire.begin();
99     Wire.beginTransmission(MPU_1);
100    Wire.write(0x6B);
101    Wire.write(0);
102    Wire.endTransmission(true);
103
104
105    Wire.begin();
106    Wire.beginTransmission(MPU_2);
107    Wire.write(0x6B);
108    Wire.write(0);
109    Wire.endTransmission(true);
110
111
112    analogWrite(11, 115); // analogWrite(11, 115);
113
114    delayMicroseconds(100);
115 }
116
117
118 void loop() {
119
120    MPU_1_acele(); // adquirir los datos del acelerometro 1
121
122    //escalo los datos para que varien entre 1g y -1g
123    double AcX1 = resolucion * AcX;
124    double AcY1 = resolucion * AcY;
125    double AcZ1 = resolucion * AcZ;
126
127
128    //calculo del angulo del acelerometro
129    double angulo1_z = Az( AcX1 , AcY1 , AcZ1 );
130    double angulo1_y = Ay( AcX1 , AcY1 , AcZ1 );
131    double angulo1_x = Ax( AcX1 , AcY1 , AcZ1 );
132
133    // Filtros pasa bajas para acelerómetro
134    angulo1_Z_fil = filtro_pasa_bajos ( angulo1_z , angulo1_Z_fil , 50.0 );
135    angulo1_Y_fil = filtro_pasa_bajos ( angulo1_y , angulo1_Y_fil , 50.0 );
136    angulo1_X_fil = filtro_pasa_bajos ( angulo1_x , angulo1_X_fil , 50.0 );
137
138    MPU_2_acele(); // adquirir los datos del acelerometro 2
139
140    //escalo los datos para que varien entre 1g y -1g

```

```

141 double AcX22 = resolucion * AcX2;
142 double AcY22 = resolucion * AcY2;
143 double AcZ22 = resolucion * AcZ2;
144
145 //calculo del angulo del acelerometro
146 double angulo1_x2 = Ax( AcX22 , AcY22 , AcZ22 );
147 double angulo1_y2 = Ay( AcX22 , AcY22 , AcZ22 );
148 double angulo1_z2 = Az( AcX22, AcY22 , AcZ22 );
149
150
151 // Filtros pasa bajas para acelerómetro
152 angulo1_Z2_fil = filtro_pasa_bajos ( angulo1_z2 , angulo1_Z2_fil , 50.0 );
153 angulo1_Y2_fil = filtro_pasa_bajos ( angulo1_y2 , angulo1_Y2_fil , 50.0 );
154 angulo1_X2_fil = filtro_pasa_bajos ( angulo1_x2 , angulo1_X2_fil , 50.0 );
155
156 float sensor_pulgar;
157 float sensor_pulgar1 = 0;
158 for (int h = 0; h < 15; h++) { // se toman 15 muestras del sensor y
159     //se realiza un promedio de esas medidas
160     sensor_pulgar = analogRead(A0);
161     delayMicroseconds(200);
162     sensor_pulgar1 = sensor_pulgar1 + sensor_pulgar;
163 }
164
165
166 sensor_pulgar = sensor_pulgar1 / 15;
167
168 //sensor_pulgar=(5.0*sensor_pulgar)/1024.0;
169 //sensor_pulgar=(180.0*sensor_pulgar)/3.5;
170
171 float sensor_indice;
172 float sensor_indice1 = 0;
173 for (int h = 0; h < 15; h++) {
174     sensor_indice = analogRead(A1);
175     delayMicroseconds(200);
176     sensor_indice1 = sensor_indice1 + sensor_indice;
177 }
178 sensor_indice = sensor_indice1 / 15; //valor que se envia a la pinza
179
180
181 //sensor_indice=(5.0*sensor_indice)/1024.0;
182 //sensor_indice=(180.0*sensor_indice)/3.2;
183
184 Serial.print((String)sensor_pulgar + '#' + (String)angulo1_X2_fil + '*' +
185     (String)angulo1_X_fil + '&' + (String)angulo1_Y_fil + '$' +
186     (String)sensor_indice + '(');
187
188 delayMicroseconds(10000);
189 /* Serial.print("Angle z: "); Serial.print(angulo1_Z_fil);
190     Serial.print("/ Angle Y: "); Serial.print(angulo1_Y_fil);
191     Serial.print("/ Angle x: "); Serial.print(angulo1_X_fil);
192     Serial.print("/ Angle z2: "); Serial.print(angulo1_Z2_fil);
193     Serial.print("/ Angle Y2: "); Serial.print(angulo1_Y2_fil);
194     Serial.print("/ Angle x2: "); Serial.println(angulo1_X2_fil);*/

```

```

195
196
197
198 char a;
199 while (Serial.available()) { //datos recibidos de la pinza
200
201     a = Serial.read();
202     R += a;
203
204     if (a == '#') { //para el sensor de fuerza 1
205         datos[0] = R.toFloat();
206         /*Serial.println("/ ");
207         Serial.println(datos[0]);
208         Serial.print("/ ");*/
209         if (datos[0] <= 147.0 && datos[0] >= 100.1 ) {
210             analogWrite(3, datos[0]);
211             delayMicroseconds(100);
212             analogWrite(5, datos[0]);
213
214
215         }
216         else {
217             analogWrite(3, 0);
218             delayMicroseconds(100);
219             analogWrite(5, 0);
220         }
221         R = "";
222     }
223     //Serial.println(datos[0]);
224
225     if (a == '*') { //para el sensor de temperatura
226         datos[1] = R.toFloat();
227         if (datos[1] >= 120.0) {
228             analogWrite(11, 255);
229             delayMicroseconds(100);
230             /*Serial.println("/ ");
231             Serial.print(datos[1]);
232             Serial.println("/ ");*/
233         }
234
235         else {
236             analogWrite(11, 115);//analogWrite(11, 115);
237             delayMicroseconds(100);
238         }
239         //Serial.print(datos[1]);
240         //Serial.println("/ ");
241         /* if (datos[1]<=147){
242             analogWrite(5,datos[1]);
243             //Serial.println("2");
244         */
245         R = "";
246         break;
247     }
248

```



## A.2 CÓDIGO IMPLEMENTADO PARA LA PINZA ROBÓTICA

```

1  #include <Servo.h>
2
3
4
5  Servo m1, m2, m3, m4, m5, m6; // variables para cada uno de los servos
6  const int pin_m1 = 2, pin_m2 = 3, pin_m3 = 4, pin_m4 = 5, pin_m5 = 6, pin_m6 = 7;
7
8  boolean flag = 0;
9  //float D1,D;
10 int datos[6]; // variable donde se van a guardar los datos resividos por el BT
11
12 String R = ""; //variable en donde se van a resivir los datos resividos por el BT
13
14 void setup() {
15     // put your setup code here, to run once:
16     // Serial2.begin(115200);
17     Serial.begin(115200);
18
19     m1.attach(pin_m1);
20     m2.attach(pin_m2);
21     m3.attach(pin_m3);
22     m4.attach(pin_m4);
23     m5.attach(pin_m5);
24     m6.attach(pin_m6);
25     /*
26         m1.write(90); // angulo de reposo 90
27         delayMicroseconds(2000);
28         m2.write(170); // angulo de reposo 170
29         delayMicroseconds(2000);
30         m2.write(90); // angulo de reposo 170
31         delayMicroseconds(2000); */
32 }
33
34 void loop() {
35
36     char a;
37     // put your main code here, to run repeatedly:
38
39     float fuerza11 = 0;
40     float fuerza22 = 0;
41     float fuerza1;
42     float fuerza2;
43     for (int h = 0; h < 10; h++) {
44
45         fuerza1 = analogRead(A5);
46         fuerza2 = analogRead(A4);
47         fuerza11 = fuerza11 + fuerza1;
48         fuerza22 = fuerza22 + fuerza2;
49     }
50     fuerza11 = 255.75 - ((fuerza11 / 10) / 4);

```

```

51 fuerza22 = 255.75 - ((fuerza22 / 10) / 4);
52
53 if (fuerza22 > fuerza11) {
54     fuerza11 = fuerza22;
55 }
56
57 if (fuerza11 > 0.9) {
58     fuerza11 = fuerza11 + 100;
59 }
60 else {
61     fuerza11 = 0.0;
62 }
63
64 /*float fuerza1 = analogRead(A5); // variable para el sensor de fuerza 1
65                                     // valores etregados 0-1023;
66     fuerza1 = 255.75 - (fuerza1 / 4.0);
67     //serial.print(fuerza1); Serial.print ("||");
68     //fuerza1=map(fuerza1, 0, 1023, 255, 0); //ya que se desea escribir este
69     //valor en voltios a la salina toca reescalar el dato de 0-255
70     float fuerza2 = analogRead(A4); // variable para el sensor de fuerza 2
71                                     //valores etrega
72     fuerza2 = 255.75 - (fuerza2 / 4.0);
73     //Serial.println(fuerza2);
74     if (fuerza1 > 0.25 || fuerza2 > 0.25) {
75         if (fuerza1 >= fuerza2) {
76             fuerza1 = fuerza1 + 100.0;
77         }
78
79         else {
80             fuerza1 = fuerza2 + 100.0;
81         }
82     }*/
83
84
85 //if (fuerza1 > 147.0){fuerza1=147.0;} //garantizo que si supera el valor
86                                     //de 147(3v) envien 147
87
88 float tem1 = analogRead(A7);
89 //tem1=(5.0*tem1)/1024.0;
90 //tem1=tem1/0.01;
91 float tem2 = analogRead(A6);
92 tem2 = -tem2 - 47;
93 //tem2=(5.0*tem2)/1024.0;
94 //tem2=tem1/0.01;
95 if (tem1 <= tem2) {
96     tem1 = tem2;
97 }
98 /*Serial.println ("||");
99     Serial.print(fuerza11); Serial.println ("||");*/
100 Serial.print((String)fuerza11 + '#' + (String)tem1 + '*');
101 delayMicroseconds(10000);
102 // Serial.print((String)fuerza1 + '#' + (String)fuerza2 + '*');
103 /*Serial.println ("||");
104     Serial.print(fuerza11); Serial.println ("||");*/

```

```

105
106
107 while (Serial.available()) { //datps recibidos del guante
108
109     a = Serial.read();
110     R += a;
111
112     //falta mirar como obtener el desplazAMIENTO PAREA el servo 1
113     if (a == '#') { //dato para el motor 1
114         //flag=1;
115         //con=con+1;
116         datos[0] = R.toFloat();
117         /*Serial.println(" ");
118         Serial.print("M1=");
119         Serial.print(datos[0]);
120         Serial.print("/ ");*/
121         //datos[0] = map(datos[0], 437, 639, 25, 160);
122         datos[0] = map(datos[0], 400, 650, 0, 180 );
123
124         if ((datos[0] % 6) == 0 && datos[0] <= 155 && datos[0] >= 20) {
125             /*Serial.println();
126             Serial.print("M1=");
127             Serial.print(datos[0]);
128             Serial.println("/ ");*/
129             m1.write(constrain(datos[0], 20 , 155));
130         }
131         //m1.write(constrain(datos[0], 30 , 150));
132
133         R = "";
134         // break;
135     }
136
137     if (a == '*') { //dato para el motor 2
138         datos[1] = R.toInt();
139         //datos[1] = datos[1];
140         datos[1] = -datos[1] + 88; //como el sensor esta ubicado ha 90 siempre le resto
141         // 90 para que cuadre es decir cuando esta en 90 muestra cero y en 0 muestra 90
142         /*Serial.println("/ ");
143         Serial.print("M2  =");
144         Serial.print(datos[1]);
145         Serial.print("/ ");*/
146         //datos[1] = map(datos[1], -90, 90, 0, 180);
147         datos[1] = map(datos[1], 16, 72, 65, 180);
148         //m2.write(constrain(datos[1], 40, 150));
149         m2.write(constrain(datos[1], 80, 150));
150
151         R = "";
152     }
153
154     if (a == '/') { //dato para el motor 3
155         datos[2] = R.toInt();
156         /*Serial.print("M3=");
157         Serial.print(datos[2]);
158         Serial.println("/ ");*/

```

```

159
160 //datos[2] = map(datos[2], 25, 0, 180, 0);
161 //datos[2] = map(datos[2], 90, -90, 180, 0);//realizo un mapeo
162
163 //m3.write(constrain(datos[2], 90, 150));//restringo los datos en los
164
165 //delayMicroseconds(10000);
166
167 R = "";
168 }
169
170 if (a == '&') { //dato para el motor 4
171   datos[3] = R.toInt();
172   /*Serial.print("M4=");
173     Serial.print(datos[3]);
174     Serial.println("/ ");*/
175   datos[3] = map(datos[3], 90, -90, 0, 180); //realizo un mapeo para
176
21
177   m4.write(constrain(datos[3], 10, 155));// restringo los datos en los valor
178
179   //delayMicroseconds(10000);
180
181   R = "";
182 }
183
184 if (a == '$') { //dato para el motor 5
185   datos[4] = R.toInt();
186   /*Serial.print("M5=");
187     Serial.print(datos[4]);
188     Serial.print("/ ");*/
189   datos[4] = map(datos[4], 90, -90, 0, 180); //realizo un mapeo para el servo 1
190   m5.write(constrain(datos[4], 20, 160));// restringo los datos en los valor
191     //de 10 a 170 grados
192   //delayMicroseconds(10000);
193
194   R = "";
195   //break; //quitar cuando ya sen envíen todo los datos (%)
196 }
197
198 if (a == '(') { //dato para el motor 6
199   datos[5] = R.toInt();
200   /*Serial.print("M6=");
201     Serial.print(datos[5]);
202     Serial.println("/ ");*/
203   datos[5] = map(datos[5], 415, 645, 15, 60);
204   m6.write(constrain(datos[5], 17, 60));
205   /*Serial.print("M6=");
206     Serial.print(datos[5]);
207     Serial.println("/ ");*/
208
209   //m6.write(17);
210
211   R = "";

```

```
212         break;
213     }
214 }
```